XVIII IMEKO WORLD CONGRESS
Metrology for a Sustainable Development
September, 17 – 22, 2006, Rio de Janeiro, Brazil

WEB SIMILARITY MEASUREMENTS USING ANT – BASED SEARCH ALGORITHM

Georgios Kouzas [1], *Eleftherios Kayafas*[2], *Vassili Loumos* [3]

[1] National Technical University of Athens, Athens, Greece, gkouzas@ece.ntua.gr
[2] National Technical University of Athens, Athens, Greece, kayafas@cs.ntua.gr
[3] National Technical University of Athens, Athens, Greece, loumos@cs.ntua.gr

Abstract: In this paper, a web search algorithm is proposed, which aims to enhance the amount of the relevant information in respect to a user's query. The proposed algorithm is based on the Ant Colony Optimization algorithm (ACO), employing in parallel document similarity issues from the field of information retrieval. Ant Colony Optimization algorithms were inspired through the observation of ant colonies. In our approach, ants are used as agents through Internet, which are capable of collecting information in each node they visit and generate routing paths through the web. The term similarity [1] is used to describe documents with contiguous content

Keywords: Web search, similarity measurements and ant agents.

1. INTRODUCTION

A rapid growth of Internet activity is observed in the last years, especially concerning WEB applications and information dissemination for many topics [2]. Unfortunately, the chaotic structure of the wild WEB makes the search of specific information at least ineffective [3]. Search engines, like Google, remarkably improved the web search but some weaknesses still remain unresolved. High percentage of irrelevant retuned results, or the reproduction of information, is very frequent in a simple query based search, in the WEB. Our system proposes an alternative way to distil and categorize the search results. Ant colony algorithms were initially used to give solutions in combinatorial problems such as the well known "Traveling Salesman Problem" [4]. However, the use of the ACO algorithms is expanded in other scientific areas like data mining [5] and, more recently, web search [6].

In our approach we suggest a modification over an ACO algorithm, which was firstly proposed by Dorigo and Maniezzo and described in reference [7]. The similarity measurement, as defined by Broder A. et all [1] and Fetterly D. et all [8], will be used for the recognition of the duplicated information.

2. DESCRIPTION OF THE ALGORITHM

The proposed algorithmic procedure is based on the following concept. An information source (web page or site) should probably lead to another information source, with a common content. Every Web page that contains relevant information is set as a starting point. Ant colony algorithms are used to direct the search from the starting point to a destination point which is another web page, linked in a close depth to the starting point. Initially, starting points are defined as the query results of a search engine like Google. The algorithm is executed for each starting point. If a web page with similar or identical content is discovered, it is defined as destination point. When a destination point is recognized, it is defined as starting point and the algorithm is repeated. The main scope of the proposed system is to group similar information. There are two main routines in the proposed algorithm, which are the ant based search algorithm and the Similarity factor.

2.1. Ant based search algorithm

The basic concept of ant colony algorithms was inspired by the observation of swarm colonies, specifically ants [9]. Since most species of ants are blind, they deposit a chemical substance called pheromone to find their way to the food source and back to their colony [10], [11]. The pheromone evaporates over time. It has been shown experimentally that the pheromone trail leads to the detection of shortest paths [12]. For example, a set of ants, initially, create a path to the food source. An obstacle with two ends is placed in their way, with one end more distant than the other. In the beginning, equal numbers of ants spread around the two ends of the obstacle. The ants, which choose the path of the nearer end of the obstacle, return before the others. The pheromone deposited on the shortest path increases more rapidly than the pheromone deposited on the longer one. Finally, as more ants use the shortest path, the pheromone of the longest path evaporates and the path disappears. In artificial life, the Ant Colony Optimization (ACO) uses artificial ants, called agents, to find solutions to difficult combinatorial optimization problems [7], [4]. ACO algorithms are based on the following concept. Each path followed by an ant is associated with a candidate solution to a given problem. The amount of pheromone deposited on a path followed by an ant is proportional to the quality of the corresponding candidate solution for the target problem.

Finally, when an ant has to choose between two or more paths, those with the larger amount of pheromone have a greater probability of being chosen by the ant.

In our approach, we propose a modification of the ACO algorithm (Dorigo M. et all [7]). In this algorithm each artificial ant employs the following properties:

- Each ant is capable of carrying memory (pheromone based)
- The node selection is based on pheromone level deposited in each node.
- Each ant has a maximum number of nodes that can visit before discovering a destination node.
- All ants start from a starting node
- Each ant uses the similarity factor to define (calculate) the document identity as described bellow.

```
Initialize system
Define Starting points: Start_List = [UMSE_Results]
Destination_List = []
Total num of Ants = NoAnts
Total number of iterations for algorithm = NoIterations
Initial pheromone value for each node added in search = IPV
Maximum number of nodes should visit each ant = Nmax
For i=0 to NoIterations
        For j=0 to NoAnts
                Init_ant
                Repeat
                        Select_Next_Node(j)
                        Query_Visited_Node(j, q′)
                        visited nodes(j)++
                Until ((visited nodes(j) = Nmax) or Query_Visited_Node(j, q′)=True)
                Calculate route(j)
                Set Visited_node to Destination_List
        End for
        Short Destination_List
        Update pheromone
End for
Set Start_List =Destination_List
```

Fig. 1. The pseudo - code of the algorithm

The figure 1 illustrates the pseudo – code of the proposed algorithm. In order to initialize our model we introduce the following parameters:

- The parameter NoA establishes the number of ants.
- An initial pheromone value equal to IPV, is set in every new linked page that is introduced in our search area.
- Each ant can visit a maximum number of nodes Nmax.
- The algorithm is repeated for a number of iterations equal to NoIterations

Let's suppose that a starting node is given by a meta-search engine. All ants are initially set to the starting point. Each time, every ant must move from node *i* to node *j*, which should be linked directly to the node *i*. The direct movement between nodes *i* and *j* is called accessibility and is described by h_{ij} parameter. If node *j* is directly linked to node *i*, the parameter h_{ij} is set to 1, otherwise it is set to zero. Let $\tau i(t)$ be the pheromone amount on node *i* at time *t*. Each ant at time *t* chooses the next node until it visits a number *N* of nodes. Therefore, we call the completion of route for each ant, an "iteration" of the Ant_Seeker algorithm. At this point the pheromone is updated according to Equation 1 where ρ is a coefficient such that *(1 - ρ)* represents the evaporation of the trail between time *t* and *t+1*, while $\Delta\tau_i$ is given according to Equation 2. In Equation 2 the quantity per unit of level of pheromone is laid on node *i* by the k^{th} ant between time *t* and *t+1* and this is expressed by Equation 3

$$\tau_i(t+1) = \rho \cdot \tau_i(t) + \Delta\tau_i \tag{1}$$

$$\Delta\tau_i = \sum_{k=1}^{m} \Delta\tau_i^k \tag{2}$$

$$\Delta\tau_i^k = \begin{cases} Q \cdot S_{MAX}^k & \text{if k ant visits node i in its tour} \\ 0 & otherwise \end{cases} \tag{3}$$

In Equation 3, *Q* is a constant and S_{MAX}^k is the maximum similarity value the ant meets on its tour. It is calculated according to Equation 7 and described to the next paragraph. The coefficient ρ must be set to a value lower than 1 to avoid unlimited accumulation of trail pheromone. An initial pheromone value equal to IPV is set when every new node is added to the search area. In order to satisfy the constraint that an ant doesn't revisit a visited node, each ant is associated with a data structure called the *vlist*, that saves the nodes already visited and forbids the ant to visit them again before a tour has been completed. When a tour is completed, the *vlist* is used to compute the ant's current solution (i.e., the node with the maximum value of Similarity factor). The *vlist* is then emptied and the ant is free to choose again.

$$P_{ij} = \frac{\tau_j \cdot h_{ij}}{\sum_{k \in allowed_k} \tau_k \cdot h_{kj}} \tag{4}$$

Where h_{ij} is the accessibility of node *j* from node *i* and is given by Equation 5.

$$h_{ij} = \begin{cases} 1 & \text{if node j is directly linked from node i} \\ 0 & otherwise \end{cases} \tag{5}$$

$$N_a = \frac{\sum_{n=1}^{Vn} (n \cdot S_n)}{\sum_{n=1}^{Vn} n} \cdot N_{\max} \tag{6}$$

The transition probability from node *i* to node *j* for the kth ant is defined at Equation 4, where *allowedk* = {Nodes can be visited - vlist}. Therefore the transition probability is a trade-off between accessibility (which states that only directly linked nodes should be chosen) and pheromone level at time *t* (which states that if this node was previously

selected then this node is highly desirable, thus implementing the autocatalytic process).

Each ant has a specific number of nodes that it can visit. This number defines the depth search of each ant. If an ant follows a path of links which contains nodes with high values of similarity, the ant has the ability to continue its search deeper. If an ant chooses a path of nodes with low values of similarity the search will stop shortly. The last visited nodes, are assigned higher priority values as far as their significance is concerned. The total number of visited nodes for each ant must not exceed a maximum value N_{max}. The expected number of nodes that each ant can visit is given by Equation 6 where Vn is the number of visited nodes and S_n is the similarity value of node n.

2.2. The Similarity Factor

The similarity factor is used to recognize the relevancy of the content. For the respective investigation we used the algorithm described in [1] and [8]. This similarity factor is based on syntactic properties of the document and is applicable to any kind of documents. In our approach the content of web pages is defined as the document.

Let's suppose that there is an N-word document. Every single word of the document is ordered to be the start of a k-word sequence. Consequently, the document is represented as a set of N-word subsequences and each subsequence is a set of *k* continuant words. Two identical documents have exactly the same set of subsequences. Two utterly different documents have no common subsequences. The similarity of two documents is defined as:

$$S_{1,2}^{k} = \frac{S_1^k \bigcap S_2^k}{S_1^k \bigcup S_2^k} \quad 0 \le S_{1,2}^{k} \le 1 \qquad (7)$$

Where S_1 is the number of subsequences appearing in the first document, S_2 is the number of subsequences appearing in the second document and *k* is the word length for every subsequence.

The *k* parameter controls the sensitivity of the similarity factor. The larger the value of the parameter *k*, the higher the sensitivity of similarity. For example, let's suppose that there are two N-word documents which differ in one single word. Each document has N subsequences and each subsequence has *k* words. Each single word appears in *k* subsequences. Thus, the number of subsequences appearing in both documents is equal to (N - *k*) and the total number of subsequences existing in both documents is (N + *k*). The value of similarity factor is (N – k)/(N + k). If the value of the parameter *k* is set equal to N then the value of similarity is equal to zero. On the other hand, if k=1 then s≈1 (N>>1), which means, that we have a word to word comparison between these documents.

3. RESULTS

We evaluated our algorithm with different training sets. Each training set is a crawled part of different web areas.

In the proposed algorithm, a set of parameters should be fine tuned. These parameters are: NoA, IPV, Nmax, ρ, NoIterations and k.

The number of ants NoA does not seem to be a critical parameter. Also the participation of the parameters IPV and ρ in the accuracy of the algorithm is very low. On the other hand, higher values of the parameters Nmax and NoIterations lead the algorithm to higher accuracy. Of course, the higher the values of these parameters the higher the execution time of the algorithm. The most crucial parameter is the value of the parameter k. Low values of the parameter k mean that a simple word comparison between two documents occurs and the algorithm cannot distinguish the relevant documents from irrelevant and finally the search is flooded with irrelevant documents. Using high values of this parameter, the algorithm fails to recognize the relevant documents. In our experiments, the value of parameter k was set to 60%.

Table 1. Results of the algorithm implementation.

	Total nodes	Relevant nodes	Percent of visits (%)	Visited nodes	Relevant nodes found	Percent found (%)
1	140594	721	64,1	90128	623	86,4
2	12329	125	35,6	4395	96	76,8
3	19553	209	64,8	12679	145	69,3
4	192155	1471	74,8	143689	1136	77,2
5	2551	38	66,2	1690	32	84.2
Average			61,1			78,8

Table 1 illustrates the results of the algorithm execution. As it is shown, the effectiveness of the algorithm is satisfying (about 79%). The different accuracy range of the algorithm in the training set is proportional to the structure of the net. The algorithm has better results in a net with high number of edges (links) per node, as it occurs in the 1st and the 5th training set. On the other hand, when the number of edges per node is low, the accuracy of the algorithm decreases, as in the 3rd training set. Another point is that the algorithm visited 61% of total pages of the training set.

4. CONCLUSIONS – FUTURE WORK

In this paper, a web search algorithm is proposed. The algorithm was inspired through the movement of real ants. Despite some weaknesses like the sensitivity in the values of parameter k, the algorithm seems to succeed in web search. Furthermore, the effectiveness of the algorithm in the large training set was remarkable. A future work is to apply the algorithm to a query based search engine. Our scope is to enhance the results of a query based search using term based document recognition instead of the similarity factor to minimize the complexity of the algorithm.

REFERENCES

[1] Broder A, Glassman S, Manasse M, Zweig G. Syntactic clustering of the Web. Proceedings of the 6th International World Wide Web Conference, April 1997; 391–404.

[2] I. Anagnostopoulos, C. Anagnostopoulos, G. Kouzas and D. Vergados, “A Generalised Regression algorithm for web page categorisation”, Neural Computing & Applications journal, Springer-Verlag, Vol. 13, no. 3, pp. 229 – 236, 2004.
[3] I. Anagnostopoulos, C. Anagnostopoulos, Vassili Loumos, Eleftherios Kayafas, “Classifying Web Pages employing a Probabilistic Neural Network Classifier”, IEE Proceedings – Software, vol. 151, no. 03, pp. 139-150, March 2004.
[4] Bianchi, L., Gambardella L.M., Dorigo M., 2002, „An ant colony optimization approach to the probabilistic travelling salesman problem”. In Proceedings of PPSN-VII, Seventh Inter17 national Conference on Parallel Problem Solving from Nature, Lecture Notes in Computer Science. Springer Verlag, Berlin, Germany.
[5] R.S. Parpinelli, et al. Data Mining with an Ant Colony Optimization Algorithm. IEEE Trans. on Evolutionary Computation, special issue on Ant Colony algorithms, 6(4), pp. 321-332, Aug. 2002.
[6] P.S. Szczepaniak et al. (Eds.): “Ants in Web Searching Process” AWIC 2005, LNAI 3528, pp. 57–62, 2005.c Springer-Verlag Berlin Heidelberg 2005
[7] Dorigo M., and Maniezzo V., 1996, “The ant system: optimization by a colony of cooperating agents”. IEEE Transactions on Systems, Man and Cybernetics, 26(1), 1-13.
[8] Dennis Fetterly, et al. “A large-scale study of the evolution of Web pages” SOFTWARE—PRACTICE AND EXPERIENCE 2004; 34:213–237
[9] Bonabeau E., Dorigo M., & Theraulaz G. “Intelligence: From Natural to Artificial Systems”, Oxford University Press.
[10] Dorigo M. and Caro G.D., 1999, “The Ant Colony Optimization Meta-heuristic,” in New Ideas in Optimization, D. Corne, M. Dorigo, and F. Glover, Eds. London: McGraw-Hill, pp. 11-32
[11] Dorigo M., and Caro G.D., 1999, “Ant Algorithms Optimization. Artificial Life”, 5(3), 137-172.
[12] Chen S., Smith. S., 1996, ”Commonality and genetic algorithms”. Technical Report CMU-RITR-96-27, The Robotic Institute, Carnegie Mellon University, Pittsburgh, PA, USA.