Ant Seeker: An algorithm for enhanced web search

Georgios Kouzas, Eleftherios Kayafas and Vassili Loumos
NTUA department of Electical, Electronic and Computer Engineering,
Multimedia Lab 11.26, Zografou Campus, 15773, Athens, Greece.
gkouzas@ece.ntua.gr, kayafas@cs.ntua. gr, loumos@cs.ntua.gr

Abstract. This paper proposes a web search algorithm, which aims to distinguish irrelevant information and 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, calculating the content similarity in each visited node and generating routing paths through the web.

1 Introduction

A rapid growth of Internet activity is observed in the last years, especially concerning web applications and information dissemination for many topics [1]. Unfortunately, the chaotic structure of the web makes the search of specific and categorized information ineffective [2]. Search engines, like Google, remarkably improved the web search but some weaknesses still remain unresolved. High percentage of irrelevant retuned results, or the information reproduction, is very frequent to a simple query based search, in the WEB. Our system proposes an alternative way to enhance information in terms of precision as well as to further categorize the search results. Ant colony algorithms were initially used to give solutions in combinatorial problems such as the well known "Traveling Salesman Problem" [3]. However, the usefulness of the ACO algorithms is expanded in other scientific areas like data mining [4] and, more recently, web search [5].

In our approach we suggest a modification over an ACO algorithm, which was firstly proposed by Dorigo and described in [6]. The similarity measurement, as defined in [7] and [8], will be used for the recognition of the duplicated information. The paper is organized as follows. The next section summarizes the related work in the field of web search in respect to the confronted problem, which must be

Please use the following format when citing this chapter:
Kouzas, Georgios, Kayafas, Eleftherios, Loumos, Vassili, 2006, in IFIP International Federation for Information Processing, Volume 204, Artificial Intelligence Applications and Innovations, eds. Maglogiannis, I., Karpouzis, K., Bramer, M., (Boston: Springer), pp. 649–656

surmounted. Section 3 describes our proposal among with all necessary algorithmic procedures and modules. In particular, we present the meta-search algorithm used for the initial result collection [9], defining in parallel, and the conception of similarity for content relevancy. Finally, we portray our modification over an ACO schema (Ant Seeker) explaining its functions analytically.

2 Related Work

With the phenomenal growth of the web, most of the search services have accepted the fact that it would be almost impossible to index the entire web. Instead, they concentrate on a specialized subset of the web and use ranking techniques to determine which of the web pages to index [10][11]. A web user is not aware of this problem and when a search service returns to him no relevant results, he will probably conclude that pertinent resources do not exist. In addition, during web search, the user must be aware of the query syntax of every search engine, a fact that renders the process even more consuming.

The evolution in the query-based web search became with the algorithm is used by Google, which looks at the links on a page and the links on pages linking to the current page. This can be used in two ways. If all the link descriptions of links to a given page could be found, then an accurate description of the given page could be created from them which do not rely on any one person's perspective or the biased perspective of a page author who includes his own meta-data. This is a potentially powerful tool in web mining [12], but to make a compendious description the whole web would have to be searched for links for the given page with obvious drawbacks. Meta-search engines became to solve the query-translation problem and the meta-results merging problems but the content-based search was still remained unsolved.

Other algorithms are focused to a content based search [4], [5], [8] and based on classification algorithms, but require a large sample set of web pages in order to be trained. Similarity factor as described in [7] and [8] provides a simple and effective approach for classifying content-relevant documents. However the content based search is still used in a small scale search.

In our approach, we are trying to combine these two different search techniques, the query based and the content based search. The first level of our search is the user filtered results of a meta-search engine in respect of a user defined query, enhanced with the second level content based search through similarity.

3 Description of the System

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 similar content. A meta-search engine collects and ranks the results of more than one search engine in order to present the results in terms of relevance. Each web page that contains relevant information is set as a starting point. Ant Seeker algorithm is used to correlate the starting point with a destination point

(another web page), linked in a close depth. In the beginning, starting points are defined as the initial query results derived from the meta-search engine. 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 reached, it is defined as a starting point and the algorithm is repeated. The similarity factor is used to assign a similarity weight in the content of different web pages. The basic functions of the proposed system are illustrated in figure 1 aiming at grouping similar information. The meta-search engine, the Similarity factor and the ant seeker algorithm are described bellow:

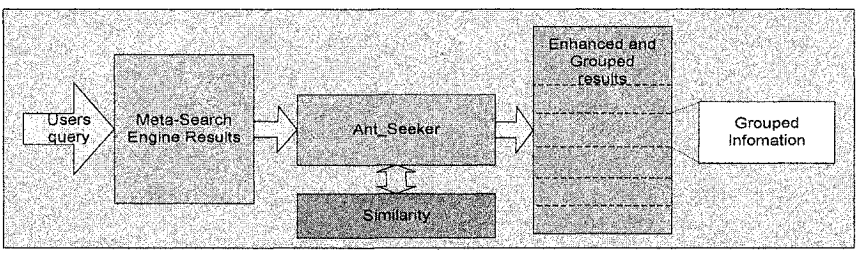

Fig. 1. The architecture of the proposed system

3.1 The meta-search algorithm

A meta-search engine is chosen for the query results instead of a typical search engine, like Google, because the meta-search engine utilizes more than one known search engine and the user gets enhanced amount of information, recording in parallel his search preferences. The meta-search engine chosen for our approach is a user-defined (UMSE) and it is described in [9]. UMSE uses a rank-based isolated merging method, since it uses information, which is readily available from search servers, without requiring any other server functionality [13],[14]. In other words the proposed method employs server-assigned ordinal ranks in order to generate the merged list of the meta-results. The UMSE 'extracts' the required information from all the submitted services combined with the meta-results and the user profile information. Then the duplicate information sources are removed. The problem of UMSE is addressed to have a search engine ranking $S = \langle R, r \rangle$, consisted of a set R of results and an ordering r. Given N ranking from N different search engines, the anticipated outcome is the generation of a single ranking $S_m = \langle R_m, r_m \rangle$, such that $R_m = R_1 \bigcup \ldots \bigcup R_N$ and r_m is the derived meta-results ranking. In other words, the merging algorithm compares whether the information source retrieved in the r^{th} rank position of search engine with priority p, exists until the $(r-1)^{th}$ rank position of the other selected search engines. The duplicate fields in the above sequence are eliminated while the procedure ends with the assignment of the last meta-result. The number of the meta-results is the total returned results from all the involved search engines, having removed the duplicated fields. UMSE allows the user to adjust the number of the returned results from each used search service. This number has a large impact on the total number and the presentation time of the meta-results.

3.2 The similarity Factor

The similarity factor is used to recognize the content relevancy. For the respective investigation, we used the algorithm described in [7] and [8]. This similarity factor is based on syntactic properties of the document. 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:

$$S_{1,2}^{k} = \frac{S_1^k \cap S_2^k}{S_1^k \cup S_2^k} \quad 0 \le S_{1,2}^{k} \le 1 \,(1)$$

Where S1 is the number of subsequences appearing in the first document, S2 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 parameter k, the bigger 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 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.

3.3 The Ant Seeker algorithm

The basic concept of ant colony algorithms was inspired by the observation of swarm colonies, specifically ants [15]. 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 [16], [17]. The pheromone evaporates over time. It has been shown experimentally that the pheromone trail leads to the detection of shortest paths [18]. 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 to the shortest path increases more rapidly than the pheromone deposited to the farther 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 [6], [3].

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.

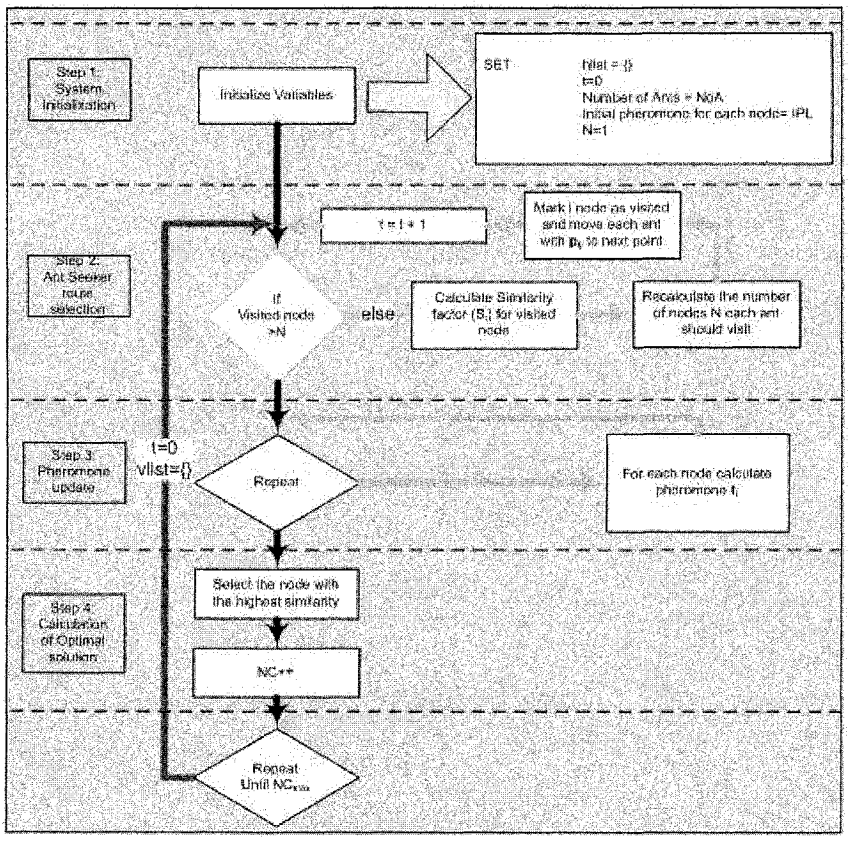

Fig. 2. Description of Ant Seeker algorithm

In our approach, we propose a modification of the ACO algorithm [6], which we call Ant_Seeker. 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 mentioned above.

The following paragraph describes how the ant seeker algorithm is applied to the web search. The figure 2 illustrates the 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 paged is introduced in our search area
- Each ant can visit a maximum number of nodes N_{max}

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 a node *i* to node *j* which should be directly linked to the node *i*. The directly movement between node *i* and *j* is called accessibility and described by h_{ij} parameter. If node *j* is directly linked to node *i*, the parameter h_{ij} is set to 1 otherwise 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 visit a number N of nodes. Therefore, we call an iteration of the Ant_Seeker algorithm the completion of route for each ant. At this point the pheromone is updated according to Equation 2, where ρ is a coefficient such that $(1 - \rho)$ represents the evaporation of trail between time *t* and $t+1$, while $\Delta\tau_i$ is given according to Equation 3. In Equation 3, $\Delta\tau_i^k$ is 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 is expressed by Equation 4.

$$\tau_i(t+1) = \rho \cdot \tau_i(t) + \Delta\tau_i \qquad (2)$$

$$\Delta\tau_i = \sum_{k=1}^{m} \Delta\tau_i^k \qquad (3)$$

$$\Delta\tau_i^k = \begin{cases} Q \cdot S_{MAX}^k & \text{if k ant visits node i in its tour} \\ 0 & \textit{otherwise} \end{cases} \qquad (4)$$

In Equation 4, Q is a constant and S_{MAX}^k is maximum similarity value the ant meets on its tour and is calculated according to Equation 1. The coefficient ρ must be set to a value lower than 1 for avoiding unlimited accumulation of trail pheromone. An initial pheromone value equal to IPV is set in every new node is added to the search area. In order to satisfy the constraint that an ant doesn't visit 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 have 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.

The transition probability from node i to node j for the k^{th} ant is defined at Equation 5, 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 highly desirable, thus implementing the autocatalytic process).

$$P_{ij} = \frac{\tau_j \cdot h_{ij}}{\sum_{k \in allowed_k} \tau_k \cdot h_{kj}} \quad (5)$$

Where h_{ij} is the accessibility of node j from node i and is given by Equation 6.

$$h_{ij} = \begin{cases} 1 & \text{if j node is directly linked from node i} \\ 0 & otherwise \end{cases} \quad (6)$$

$$N_a = \frac{\sum_{n=1}^{Vn} (n \cdot S_n)}{\sum_{n=1}^{Vn} n} \cdot N_{\max} \quad (7)$$

Each ant has a specific number of nodes that 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 with higher weights 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 7 where Vn is the number of visited nodes and S_n is the similarity value of node n.

4 Conclusion

It is concluded that sustainable development for web search may be achieved with the conjunction of well known optimization algorithms and similarity metrics. The final assessment of the proposed method will be evaluated on a large set of web pages.

In addition, we undertake a research among other Artificial Intelligence methods in order to create a hybrid approach to provide an automatically adjustment of the number of the returned results from each search service used. The main scope is to achieve a successful combination of "query-based" and "semantic-based" operation in our system.

References

1 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.

2 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.

3 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.

4 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.

5 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

6 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.

7 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.

8 Dennis Fetterly, et al. "A large-scale study of the evolution of Web pages" SOFTWARE—PRACTICE AND EXPERIENCE 2004; 34:213–237

9 Anagnostopoulos I., Psoroulas I., Loumos V. and Kayafas E., "Implementing a customized meta-search interface for user query personalization", , 24th International Conference on In-formation Technology Interfaces, ITI 2002, pp. 79-84, June 24-27, 2002, Cavtat/Dubrovnik, CROATIA.

10 Oyama S, Kokubo T, Ishida T (2004) Domain-specific Web search with keyword spices. IEEE Transactions on Knowledge and Data Engineering. 16(1):17–27

11 Pokorny J (2004) Web searching and information retrieval. Computing in Science & Engineering. 6(4):43-48

12 Soumen Chakrabartia, Byron Doma, Prabhakar Raghavana, Sridhar Rajagopalana, David Gibsonb, and Jon KleinbergcAutomatic. Automatic Resource compilation by analyzing hyperlink structure and associated text, 1998.

13 Craswell, Nick, Hawking, David and Thistlewaite, Paul. Merging Results from Isolated Search Engines. 10th Australasian Database Conference, Auckland, New Zealand, January 1999, Springer-Verlag, Singapore.

14 Yuwono, Budi and Lee, Dik L. Server ranking for distributed text retrieval systems on the internet. In Topor, Rodney and Tanaka, Katsumi, editors, DASFAA '97, pages 41-49, Melbourne. World Scientific, Singapore.

15 Bonabeau E., Dorigo M., & Theraulaz G. "Intelligence: From Natural to Artificial Systems", Oxford University Press.

16 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.

17 Dorigo M., and Caro G.D., 1999, "Ant Algorithms Optimization. Artificial Life", 5(3), 137-172.

18 Chen S., Smith. S., 1996, "Commonality and genetic algorithms". Technical Report CMU-RITR-96-27, The Robotic Institute, Carnegie Mellon University, Pittsburgh, PA, USA.