



ANT COLONY AND CUCKOO SEARCH ALGORITHMS FOR DOCUMENT CLUSTERING

Priya Vijayanthi

Department of Computer Science and Engineering, GMR Institute of Technology, Andhra Pradesh, India

E-Mail: priyavijayanthi.r@gmr.it.org

ABSTRACT

One of the ways to improve the efficiency of Information Retrieval (IR) systems is through document clustering. The search result of an IR system can be grouped or clustered so that the retrieval is made faster. Efficiency of IR systems has to be improved without compromising the quality of clusters. This paper presents a comparative study of the quality of cluster results by solving the problem of document clustering using Ant Colony Optimization (ACO) algorithm and Cuckoo Search Optimization (CSO) algorithms. The results of experiments done on Classic 4 dataset shows that both the algorithms are equally good in giving good quality clusters. The quality of clusters is measured in terms of F-measure and DB-index.

Keywords: document clustering, ant colony, cuckoo search, optimization, information retrieval.

1. INTRODUCTION

Information Retrieval (IR) is an emerging subfield of information science concerned with the representation, storage, access, and retrieval of information. Current research areas within this field are searching and querying, ranking search results, navigating and browsing information, optimizing information representation and storage, document classification and document Clustering. The amount of data that is added to world wide web is exponentially increasing every day and every minute and second. Web is expanding exponentially and Web Technologies have introduced problems of their own. Searching and finding useful information is increasingly becoming a hit-or-miss practice that often ends in information overload. Search engines are common gateways to the huge collections of electronic text on the Web. They are continuously optimized and enhanced to serve their users in a better way. It has been accepted that the precision of the search results are low. Precision is measured in terms of the number of relevant documents in the search result compared to the total number of documents in the result. Rueger and Ganch (2000) outlined three approaches to amend information retrieval system. One among them is to cluster the documents that allow the users to better preview and navigate the information structure of the returned results. Clustering is the process of dividing a set of objects into several groups or clusters in such a way that the objects placed in a group are highly similar and the objects placed in different groups are highly dissimilar.

The number of different ways in which a set of objects can be grouped is a factorial function. This makes the problem of document clustering as combinatorial optimization problem and so many researchers have used popular optimization techniques.

Recently, the interest in the application of nature inspired algorithms such as Ant Colony Optimization (ACO) algorithm and Cuckoo Search algorithms (CSO) has grown due to various reasons such as the generation of population of solutions (instead of manipulating a single solution) and explicit memory of previously visited

solutions. In general ACO and CSO possesses several distinctive features in terms of robustness, flexibility and distributives which lead to solve a wide spectrum of problems like Travelling Salesman Problem (Dorigo and Gambardella 1997a,b), Vehicle Routing Problem (Bullnheimer *et al* 1997), Quadratic Assignment Problem (Gambardella *et al* 1999) and Graph Colouring Problem (Costa and Hertz 1997). ACO and CSO are most successful examples of swarm intelligent systems and are expected to provide promising solutions in the area of document clustering.

Bo Liu *et al* (2003) have proposed an improved Ant-Miner algorithm called Ant_Miner3 for classification rule discovery in database. The pheromone updating rules proposed in previous versions of Ant-Miner algorithms were compared. In the proposed algorithm, new pheromone update method was introduced. They proved that the performance of the algorithm is better than Ant-Miner by testing it on Wisconsin breast cancer database and Tic-tac-toe endgame database. The authors concluded that the new pheromone updating method used and the new state transition rule used increase the accuracy of the classification by ACO. Cheng-Fa *et al* (2002) have proposed a new data clustering method for data mining in large databases. The proposed method Ant Colony Optimization with different favour ACODF is based on ACO. It uses differently favourable ants to solve clustering problem and adopts simulated annealing to decrease the possibility of getting local optimal solutions. Furthermore, it utilizes tournament selection strategy to choose a path. The algorithm is tested with 400 data sets. It has been shown that the performance of ACODF algorithm is better than Fast SOM combined K-means and Genetic K-means algorithm (GKA).

Wai-chiu *et al* (2000) have proposed a new feature extraction mechanism for Web page classification and a tree structure called the DC tree to make the clustering process incremental and less sensitive to the document insertion order. The proposed algorithm uses the coverage factor for the selection of features to represent the documents. The coverage of the feature is defined as



the percentage of documents containing at least one feature of the extracted features. This method does not depend on the term frequency. The DC tree proposed is a tree in which each node represents a document cluster. It has been proved that the proposed techniques classify the web documents more effectively. Ramos *et al* (2002) have proposed some models derived from the observation of real ants, emphasizing the role played by stigmergy as distributed communication paradigm. Based on the models derived, the authors propose a new ant clustering system called ACLUSTER. It is stated by the authors that the proposed algorithm avoids not only short term memory based strategies, as well as the use of several artificial ant types. The proposed ACLUSTER algorithm is used for self-organized data and image retrieval. The authors stated that ACLUSTER algorithm can be used to tackle unsupervised data exploratory analysis as well as data retrieval systems.

Thangavel *et al* (2007) have proposed an enhanced ant colony optimization algorithm for mining classification rule called Threshold Ant Colony Optimization (TACO-Miner). The aim of the proposed algorithm is to provide comprehensible classification rules that have higher predictive accuracy and simpler rule list. In the proposed algorithm, a new procedure is introduced in the rule construction. The information gain value is used to check the credibility of the term that is being selected to present in the rule. The experimental results were compared with that produced by Ant-Miner, ACO-Miner and C4.5. It has been proved by the authors that the results produced by the TACO-Miner have a higher predictive accuracy. Also the experiments conducted help to understand the influence of system parameters on the performance of the algorithm. Wu *et al* (2002) have proposed a document clustering method based on swarm intelligence and K-means algorithm. The proposed CSIM algorithm first employs a document clustering algorithm based on swarm intelligence. It is derived from a basic model of interpreting the ant colony organization from cemeteries. Then the traditional K-means clustering algorithm is used. The authors state that the proposed hybrid algorithm hides the limitations of the two techniques combined. It has been proved that the hybrid CSIM algorithm gives better performance.

Yang *et al* (2005) have proposed new algorithm using ant colony algorithm and validity index for document clustering. In order to reduce the outliers, the algorithm uses validity index to find the optimal number of clusters. The experiments conducted on Reuters-21578 collection proved that the proposed algorithm performs better than ART neural networks. Julia *et al* (2008) have proposed ant based and swarm based clustering algorithm for document clustering. It has been shown that the proposed algorithm mimics the behaviour of real ant colonies. An analysis on the usage of swarm based algorithms for document clustering is given and it is stated that the ant based algorithms are the most widely used swarm based algorithms. Sara *et al* (2005) have proposed a novel concept of ACO and its learning mechanism integrated with K-means algorithm to solve image

clustering problem. It has been proved that the undesired solutions of K-means algorithm are omitted by the learning mechanism found with ant colony algorithm. Shang *et al* (2004) have proposed a new algorithm namely DBAnt Cluster which is used with Ant Class algorithm to improve its performance. In the proposed method, high density clusters are obtained using DBSCAN algorithm and the resultant clusters are scattered on a grid board. It has been shown that there was improvement in the performance of Ant Class algorithm. Chen *et al* (2004) have proposed a new heuristic density based ant colony clustering algorithm (HDACC). The algorithm uses a memory bank to store the heuristic knowledge which guides the ants. The experimental results prove that the proposed HDACC algorithm is a viable and effective clustering algorithm.

Kuo *et al* (2005) have proposed a novel clustering method, ant K-means (AK) algorithm. It modifies the K-means algorithm by locating the objects in a cluster with the probability which is updated by the pheromone and the rule for updating the pheromone is according to total within cluster variance (TWCV). The authors have shown that the algorithm gives better performance when compared to the results produced by self-organizing map followed by K-means and SOM followed by genetic K-means algorithm. Mohammad *et al* (2007) have proposed a new population based search algorithm called Bees algorithm which is capable of locating near optimal solutions efficiently. The proposed HBMK-means algorithm combines the new Bees algorithm with the K-means algorithm. A colony of honey bees can extend itself over long distances in order to exploit a large number of food sources at the same time. The proposed algorithm exploits the search capability of the Bees to overcome the local optimum problem of the K-means algorithm. The Euclidean distance measure is used to determine the closeness between a pair of documents. The authors have tested the proposed algorithm over five real data sets which are benchmark data sets and the results were compared with that of K-means algorithm and GA. It has been proved that the results produced by Bees algorithm are better than that of K-means algorithm and GA. Taher *et al* (2009) have proposed a hybrid evolutionary programming based clustering algorithm called PSO-SA by combining PSO with simulated annealing. The proposed algorithm makes the search around the global solution through SA and increases the information exchange between the particles using a mutation operator to escape local optima. Cui *et al* (2005) have proposed a method that uses PSO for clustering documents. It has been observed that PSO algorithm performs a globalized search in the entire solution space. The PSO algorithm is hybridized with K-means algorithm. The whole clustering behaviour of the PSO clustering algorithm is divided into two stages, a global searching stage and a local refining stage. It is stated by the authors that if given enough time, the PSO algorithm could generate more compact clustering results from a low dimensional data set than K-means clustering algorithm. When clustering large document data sets, slow shift from



global searching stage to local refining stage causes PSO algorithm to take more iterations to converge to the optima in the refining stage than required by K-means algorithm. It is illustrated through experimental results that hybrid PSO algorithm can generate higher compact clustering than either the PSO or the K-means alone. Ingaramo *et al* (2009) have proposed some variants of PSO for clustering short text collection. They used different representations for the very short text collections and used two unsupervised measures of cluster validity namely, Expected Density Measure and Global Silhouette coefficient. It has been shown that PSO yields better results and outperforms other techniques. Yang and Deb (2010) have proposed a new metaheuristic optimization algorithm called Cuckoo Search (CS) algorithm. CS algorithm is based on the breeding strategy observed with some species of cuckoos. The authors have proved the viability of CS algorithm by testing with benchmark functions. It has been shown that the CS algorithm outperforms PSO algorithm. Yang and Deb (2009) have formulated and proposed a new metaheuristic algorithm called, Cuckoo Search (CS) algorithm. It is based on the observation that the flight performed by certain species of cuckoos follow Levy distribution. In CS algorithm Levy flight is performed to generate new solutions. It has been shown by the authors that CS algorithm surpass GA and PSO algorithms.

2. RESEARCH METHOD

The problem of document clustering is formulated as an unconstrained combinatorial problem. The problem is to partition the given document collection D into specified number of clusters. Let there are 'n' documents in D . Let the number of clusters be 'k'. Then, the problem is to partition 'n' documents into 'k' clusters and clustering method used is Partition based hard clustering. The number of different ways in which 'n' document can be clustered into 'k' clusters is formulated as an optimization problem. The reason is that, the number of possible ways in which 'n' objects can be divided into 'k' groups follows Stirling Number as given in Equation (1).

$$SN = \frac{1}{k!} \sum_{i=1}^k (-1)^{k-i} \left(\frac{k}{i} \right) i^n \quad (1)$$

It is clear that with smaller values for 'n' and 'k' as 10 and 3 respectively, the number of possible partitions comes to 9330. Examining all possible partitions and identifying the global optimum partitions are not computationally feasible. In the present work, vector space model is used. Pre-processing is done before representation. Pre-processing includes stop word removal, stemming and unique word identification. A unique set of words from all the documents of the document corpus 'D' or document collection is obtained and this is used to represent each document in the document collection. The set of terms or words obtained in

the corpus is weighted. A document d_i is represented as a vector as per Equation (2). Each element in the vector is the weight of each term in the corpus.

$$d_i = \{w_{i1}, w_{i2} \dots w_{im}\} \quad (2)$$

In the above equation, any element say, w_{ij} is the weight of term 'j' in ' d_i ' and 'm' represent the total number of terms in the corpus. The weight of each word or term represents the importance of the word in the document. It is calculated using Equation (3).

$$w_{ij} = tf_j * idf_j \quad (3)$$

In this equation, w_{ij} is the weight of term j in d_i , tf_j is the frequency of term j in d_i , idf_j is the inverse document frequency. Inverse document frequency is based on the importance of term j in other documents in the document corpus. The solution to the problem of document clustering is a vector where each component corresponds to the centre of a cluster. The solution is represented as in Equation (4).

$$C = (c_1, c_2, c_3 \dots c_k) \text{ where} \quad (4)$$

'C' is the solution vector and c_i is the centroid of each cluster.

The similarity measure used is the Cosine similarity because it best reflects the similarity between text documents. The Cosine similarity between a pair of vectors is given by Equation (5).

$$\cos = \frac{A \cdot B}{\|A\| \|B\|} \quad (5)$$

Here, A and B are vectors of same length.

For clustering, two measures of cluster goodness or quality are used. One type of measure compares different sets of clusters without reference to external knowledge and is called as Internal Quality Measure. The overall similarity based on the pair wise similarity of documents in a cluster can be used. The other type of measure evaluates how well the clustering is working by comparing the groups produced by the clustering techniques to known classes. This type of measure is called as External Quality Measure.

The quality of the solution vector is measured by Davis-Bouldin (DB) index as in Equation (6) which is based on internal criterion.

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{i \neq j} \left(\frac{\sigma_i + \sigma_j}{\text{dist}(c_i, c_j)} \right) \quad (6)$$



where, 'k' is the number of clusters, σ_i is the average distance of all the documents in cluster 'i' to its centre ' c_i ' and $\text{dist}(c_i, c_j)$ is the similarity between c_i and c_j . In Equation (6), as the similarity between the documents is to be maximized and for good quality clusters the value in the denominator should be larger. Also, if the clusters are compact, then the average distance of all the documents in a cluster should be decreased. For a good quality cluster, the value of the numerator term should be smaller. Thus, for a good quality cluster, the value of DB-Index is lower. In other words, lower the value of DB-Index, higher is the quality of cluster.

An external quality measure called F-measure is also used. It combines the Precision and Recall ideas from Information Retrieval. Each cluster is treated as a result of a query and each cluster is the desired set of documents for a query. First recall and precision of the cluster for each known class are calculated. More specifically, for each cluster j and class i,

$$\text{Recall}(i, j) = \frac{n_{ij}}{n_i} \quad (7)$$

$$\text{Precision}(i, j) = \frac{n_{ij}}{n_j} \quad (8)$$

where n_{ij} is the number of members of class i in cluster j, n_i is the number of members of class i and n_j is the number of members of cluster j.

Using Equations (7) and (8), F-measure can be calculated as given in Equation (9).

$$F(i, j) = \frac{(2 * \text{Recall}(i, j) * \text{Precision}(i, j))}{(\text{Precision}(i, j) + \text{Recall}(i, j))} \quad (9)$$

F-measure of the whole clustering is given by

$$F = \sum_i \frac{n_i}{n} \max\{F(i, j)\} \quad (10)$$

The higher the value of F-measure, higher is the quality of cluster.

2.1 ACO algorithm

The basic idea of ACO algorithm is to use a positive feedback mechanism, based on an analogy of trail-laying and trail-following behaviour of the real world ant species. This process of trail-laying reinforces the portions of good solutions that contribute to the quality of these solutions. A virtual pheromone, used as reinforcement, allows good solutions to be kept in memory, from where they can be used to make up better solutions. The problem of document clustering is formulated in the same way as Travelling Salesman

Problem (TSP). Each document in the document corpus is treated as a node. The amount of pheromone deposition between two documents represents the edge and is proportional to the similarity between the documents. Each virtual ant in the colony constructs a graph connecting all the documents in the corpus. The graph is disconnected using graph algorithms like minimum spanning tree which gives the resultant clusters of documents. In the process of graph construction, an ant at document d_i moves to document d_j if the similarity between d_i and d_j is more and if d_j is not yet visited by it. On each move from d_i to d_j at time t, each virtual ant deposits a small amount of pheromone and is denoted by $\delta_{ij}(t)$ and it is proportional to the similarity between d_i and d_j and it follows Equations (11 and 12).

$$\delta_{ij}(t+1) = \delta_{ij}(t) (1-\rho) + \Delta\delta \quad \text{where} \quad (11)$$

$$\Delta\delta = \begin{cases} \sum_{j=1}^{N_i} \left[1 - \frac{\text{dist}(c_i, d_j)}{\gamma} \right] & d_j \in c_i \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

Equation (11) represents the amount of pheromone deposited on the edge connecting the documents d_i and d_j at time (t+1). The first term of Equation (3.1) is the product of pheromone deposition that exists already in the edge and pheromone decay. This is introduced to mimic the exact behaviour of real world ant species. The second term of Equation (11) is the fractional amount of pheromone deposited by a virtual ant on the edge between d_i and d_j through its move. The fractional amount of pheromone deposited by an ant is based on the integrated similarity of a document with other documents within a cluster and is given by Equation (12). The distance function used in Equation (12) is the cosine similarity between the center of cluster i and the document d_j . c_i is the cluster center of cluster i and is a vector, N_i is the number of documents in cluster i and the parameter γ is defined as the swarm similarity coefficient and it influences the convergence of the algorithm.

The move by a virtual ant from d_i and d_j is based on the probabilistic state transition rule given in Equation (13). The parameter ψ is a problem-dependent heuristic function for the document pair d_i and d_j .

$$P_{ij}^a(t) = \begin{cases} \frac{[\delta_{ij}(t)]^g [\psi_{ij}(t)]^h}{\sum_{j \notin \text{Tabu}_a} [\delta_{ij}(t)]^g [\psi_{ij}(t)]^h}, & \text{if } j \notin \text{Tabu}_a \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

$P_{ij}^a(t)$ represents the probability with which the virtual ant moves from document d_i to document d_j at time t. Here g and h are parameters to control trail intensity. As already discussed, Ψ is the problem dependent heuristic



function, and here it is taken to be the similarity between the documents d_i and d_j . The pseudo code for the ACO

algorithm is given Figure-1.

Pseudo code for ACO algorithm

1. *Initialization*
 set iteration counter to 0
 for every edge (i, j) between documents ' i ' and ' j ',
 initialize the trail intensity
2. Randomly generate and place the ants to begin their tour.
3. Add starting document to the tabu list of the corresponding ant
4. *Graph construction*
 Repeat until the tabu list of all ants is full
 for each ant $a = 1$ to v_a do
 Select document ' j ' to move from document ' i ' with
 probability $P_{ij}(t)$
 Place document ' j ' in tabu list of ant ' a '
 Move ant ' a ' to document ' j '
 end for
 end repeat
5. *Pheromoneupdateation*
 for every edge (i, j) do
 $\Delta\delta_{ij} = \sum_{a=1}^{v_a} \Delta\delta_{a,ij}$
 compute $\delta_{ij}(t+1) = (1 - \rho) \delta_{ij}(t) + \Delta\delta_{ij}$
 set $\Delta\delta_{ij} = 0$
 end for
6. if stopping criterion is not met
 continue from step 2
 else
 Give the current best solution and stop

Figure-1. Pseudo code for ACO algorithm.

2.2 CSO algorithm

The basic version of CSO algorithm uses the following representations. Each egg in a nest represents a solution, and a cuckoo egg represents a new solution. The aim is to use the new and potentially better solutions (cuckoos) to replace the one not-so-good solution in the nests. In the simple form, each nest has only one egg. The algorithm can be extended to more complicated cases in which each nest can have multiple eggs representing a set of solutions. The basic algorithm uses the following rules.

- a) Each cuckoo lays one egg at a time, and dumps its egg in a randomly chosen nest.

- b) The best nests with high quality of eggs will be carried over to the next generation.
- c) The number of available hosts is fixed and the egg laid by a cuckoo is discovered by the host bird with a probability
 $p_a \in (0,1)$.

Based on these rules, the basic steps of the CSO algorithm is given in pseudo code as shown in Figure-2.



Pseudo code for CSO
Randomly generate solutions and place in each nest Find the best solution in each nest Repeat for each cuckoo do Generate a cuckoo egg (new solution) by performing Levy flight Randomly choose a nest If the quality of cuckoo egg is better than best in chosen nest Replace worst in nest with cuckoo egg end for Update the current best solution Discard nest with poor quality solutions with discard probability and generate new nest until stopping criterion not met

Figure-2. Pseudo code for CSO algorithm.

The new solution is generated as per Equation (14) and a new solution $x^{(t+1)}$ is generated by a cuckoo by performing a Levy flight in the current solution.

$$x^{(t+1)} = x^{(t)} + \alpha \oplus \text{Levy}(\lambda) \quad (14)$$

where $\alpha > 0$ is related to the scales of the problem of interests. Generally α is taken as 1. This equation is a stochastic equation for random walk. In general, a random walk is a Markov chain whose next status depends on the current location and the transition probability. The symbol \oplus represents the entry-wise multiplication. In basic version of CSO algorithm, the random walk is Levy flight and so the algorithm is more efficient in exploring the search space. This is because of the fact that the step size is longer in long run. The step length is drawn from Levy distribution as given in Equation (15) using Mantegna's algorithm.

$$\text{Levy} \sim u = t^{-\lambda}, (1 < \lambda \leq 3) \quad (15)$$

where u is drawn from normal distribution. The Levy flight provides a random walk and this forms a random walk process with a power-law step length distribution with a heavy tail. New solutions are generated around the current best solution and this helps to speed the local search. But, significant number of new solutions is generated by far field randomization and location of these

solutions is far enough from the current best solution. This guarantees that the algorithm will never trap at local optima.

3. RESULTS AND ANALYSIS

The feasibility of the ACO algorithm for the problem of document clustering is thoroughly tested with a standard document corpus. Several experiments are conducted by selecting subset of documents from the standard benchmark datasets namely, LISA and Classic4. The quality of solutions generated by ACO algorithm is evaluated in terms of cluster quality through an internal quality measure namely, DB- Index and through an external quality measure namely, F-measure. Initially numerical experiments are conducted on a set of 100 and 200 documents selected from LISA which consists of 635 documents to ascertain the most beneficial values of certain parameters like g , h , ρ and $\delta(t)$.

The values of g , h , ρ and $\delta(t)$ involved in ACO are taken as 1, 1, 0.1 and 0.4 respectively. The numerical experiments are repeated by varying the number of iterations from 100 to 500. The number of ants (v_a) is varied from 4 to 10 with an increment of 2 and the number of clusters k is taken as 4. The same experiments were done for CSO algorithm where the number of nests is taken to be equal to the number of clusters. The number of solutions in each nest is taken as 5 and the step length is taken as 1. The solutions generated by CSO are compared with that of ACO algorithm.

**Table-1.** Cluster Quality: CSO Vs ACO for LISA.

No. of iterations	n = 100		n = 200	
	CSO	ACO	CSO	ACO
100	0.5945	0.6208	0.6123	0.6021
200	0.6219	0.6391	0.6072	0.5981
300	0.6309	0.6022	0.6274	0.6109
400	0.6197	0.6293	0.6108	0.6185
500	0.6141	0.6198	0.6128	0.6119
Mean	0.6162	0.6222	0.6141	0.6083

It is clear from the results furnished in Tables 1 that for a smaller document set, the performance of CSO algorithm is same as that of ACO algorithm.

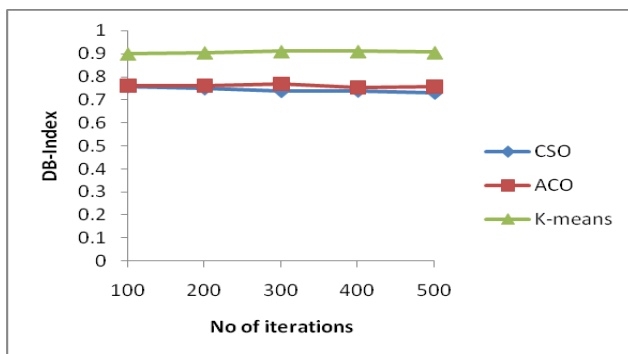
The same kind of experiment is conducted to test the performance of ACO and CSO algorithms by varying

the number of documents from 600 to 2000 taken from Classic4 dataset. The experiments are conducted for n=600, n=800, n=1200 and n=2000. The results are tabulated in Table-2.

Table-2. Cluster quality: CSO Vs ACO for Classic4.

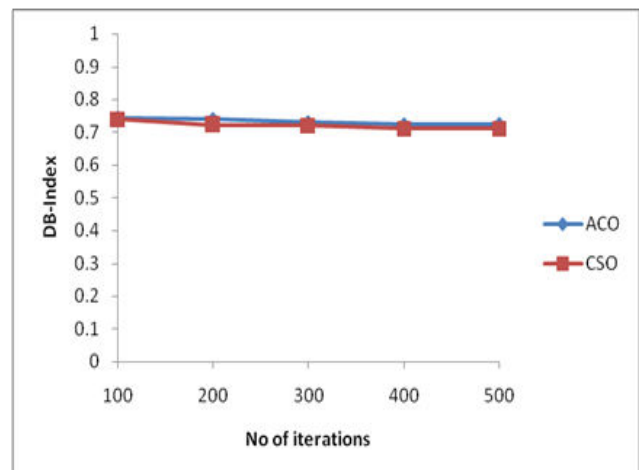
No. of iterations	n = 600		n = 800		n=1200		n=2000	
	CSO	ACO	CSO	ACO	CSO	ACO	CSO	ACO
100	0.6192	0.6589	0.6912	0.6891	0.7109	0.7549	0.7581	0.7603
200	0.6205	0.6408	0.6895	0.7193	0.7083	0.7528	0.7494	0.7612
300	0.6163	0.6502	0.6810	0.7134	0.7134	0.7491	0.7381	0.7681
400	0.6286	0.6539	0.6791	0.7119	0.7093	0.7502	0.7393	0.7538
500	0.6117	0.6542	0.6735	0.7058	0.7104	0.7488	0.7307	0.7558
Mean	0.6193	0.6516	0.6829	0.7079	0.7105	0.7512	0.7431	0.7598

Following graph compares the quality of clusters obtained from ACO, CSO and standard K-means algorithms for n=2000;

**Figure-3.** Cluster Quality: CSO Vs ACO, K-means n=2000.

The similar kinds of experiments are conducted by varying the number of documents from 2500 to 5000 with an increment of 500 documents in each experiment. The results obtained using CSO algorithm are compared with that of ACO and are shown in Figures 4 - 9. It is evident from the results that CSO algorithm gives a better performance when compared to ACO algorithm. However

the improvement is marginal. The percentage of improvement in cluster quality is examined for CSO algorithm and is given in Figure-10. This depicts that CSO algorithm can be employed for large datasets.

**Figure-4.** Cluster Quality: CSO Vs ACO for n=2500.

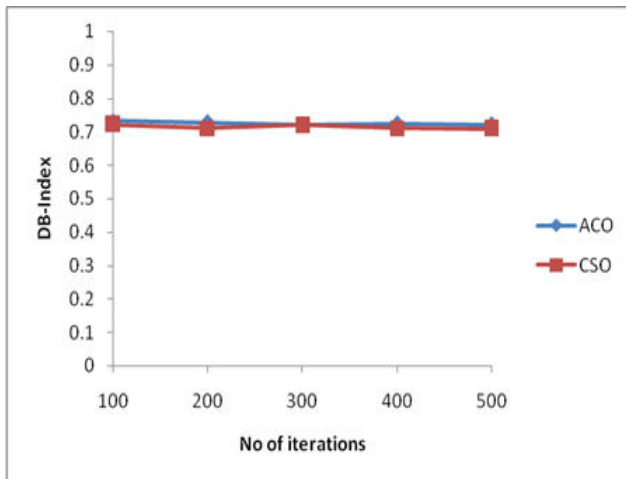


Figure-5. Cluster Quality: CSO Vs ACO for n=3000.

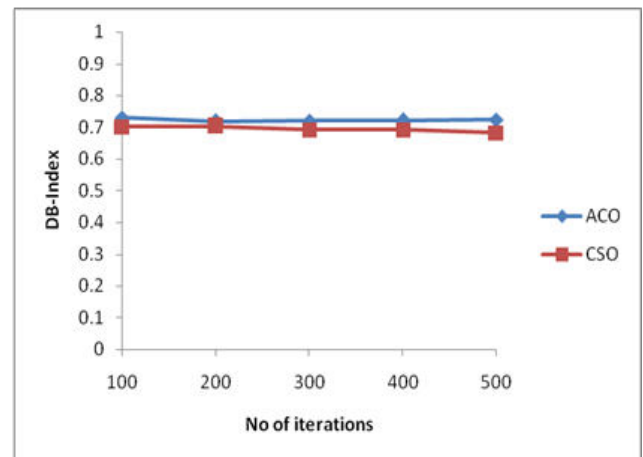


Figure-8. Cluster Quality: CSO Vs ACO for n=4500.

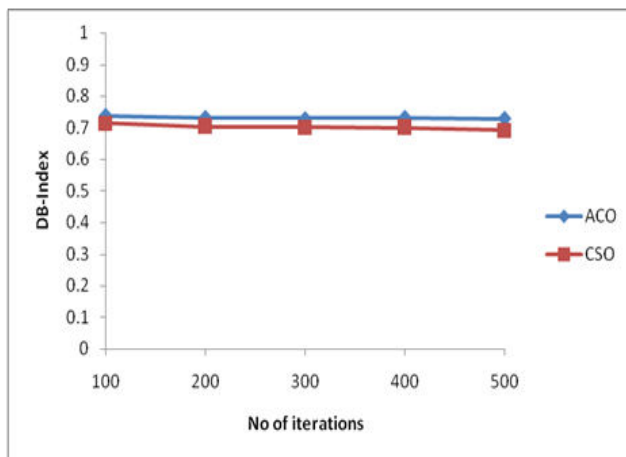


Figure-6. Cluster Quality: CSO Vs ACO for n=3500.

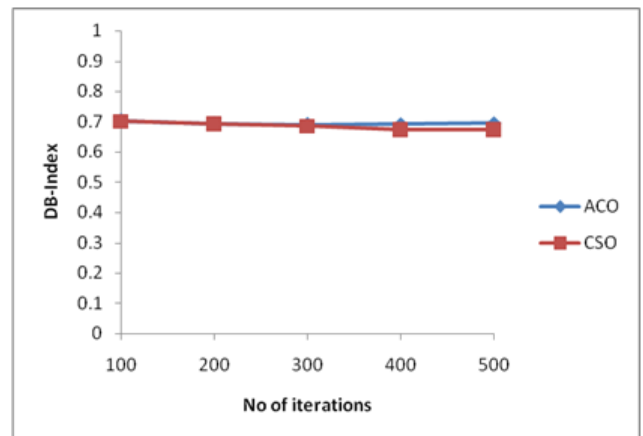


Figure-9. Cluster Quality: CSO Vs ACO for n=5000.

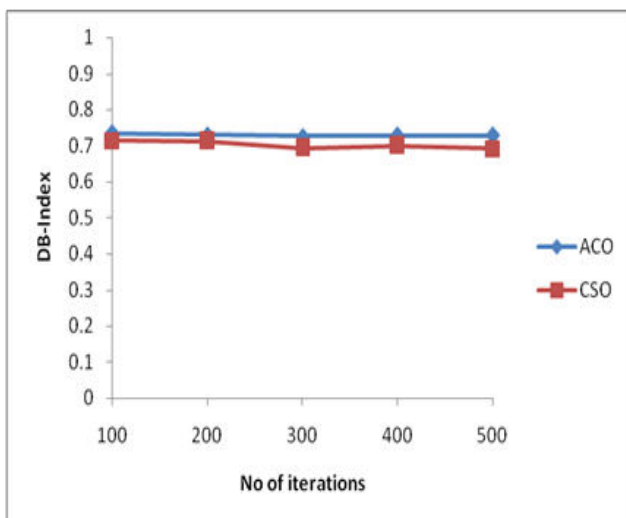


Figure-7. Cluster Quality: CSO Vs ACO for n=4000.

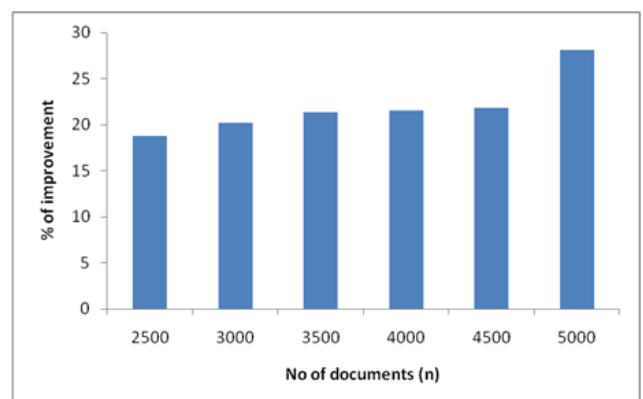


Figure-10. % of improvement in cluster quality using CSO, ACO over K-means.

4. CONCLUSIONS

The problem of document clustering is addressed using 2 popular bio-inspired algorithms ACO and CSO. The viability of the algorithms is tested by evaluating the quality of clusters. For smaller datasets, the performance of both ACO and CSO algorithms are almost same. However, CSO algorithm is found to perform better when dataset is large. It is suggested that the performance of the bio-inspired algorithms can be improved by combining



other search methods like Tabu-search to order to increase the reachability of several regions of solution space. Also, it is suggested that blended forms of ACO and CSO algorithms can be experimented with very large datasets.

REFERENCES

- Yang X. S. *et al.* 2010. Engineering Optimization by Cuckoo Search. *International Journal of Mathematical Modelling and Numerical Optimization*. 1(4): 330-343.
- Yang X. S. *et al.* 2009. Cuckoo search via Levy flights. in the Proceedings of World Congress on Nature and Biologically Inspired Computing (NaBIC 2009), December 2009, India. IEEE Publications, USA. pp. 210-214.
- Taher N., *et al.* 2009. An Efficient Hybrid Evolutionary Optimization Algorithm based on PSO and SA for Clustering. *Journal of Shejiang University - Science A*. 10(4): 512-519.
- Julia H. *et al.* 2008. Ant based and Swarm based Clustering. *LNCS on Swarm Intelligence*. 1(2): 95-113.
- Mohammad F., *et al.* 2007. Application of Honey Bee Mating Optimization Algorithm on Clustering. *Applied Mathematics and Computation*. 190(2): 1502-1513.
- Thangavel K., *et al.* 2007. Rule Mining Algorithm with a New Ant Colony Optimization Algorithm. *International Conference on Computational Intelligence and Multimedia Applications*. pp. 135-140.
- Cui X. *et al.* 2005. Document Clustering using Particle Swarm Optimization. *Proceedings of IEEE Swarm Intelligence Symposium*. pp. 185-191.
- Sara S. *et al.* 2005. Hybridization of the Ant Colony Optimization with the K-means Algorithm for Clustering. *LNCS on Image Analysis*. 3540: 283-293.
- Yang Y., *et al.* 2005. A Model of Document Clustering using Ant Colony Algorithm and Validity Index. *Proceedings of 2005 IEEE International Joint Conference on Neural Networks*. pp. 2730-2735.
- Shang L., *et al.* 2004. A New Ant Colony Algorithm based on DBSCAN. *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*. pp. 1491-1496.
- Chen *et al.* 2004. HDACC: A New Heuristic Density based Ant Colony Clustering Algorithm. *Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology*. pp. 397-400.
- Bo Liu, *et al.* 2003. Classification Rule Discovery with an Ant Colony Optimization. *Proceedings of IEEE/WIC International Conference on Intelligent Agent Technology*. pp. 83-88.
- Ramos V., *et al.* 2002. Self-Organized Data and Image Retrieval as a Consequence of Inter-Dynamic Synergistic Relationships in Artificial Ant Colonies. *Hybrid Intelligent Systems, IOS Press, Frontiers of Artificial Intelligence and Applications*. Vol. 87.
- Wu B., *et al.* 2002. CSIM: a Document Clustering Algorithm based on Swarm Intelligence. *Proceedings of 2002 Congress on Evolutionary Computation*. pp. 477-482.
- Wai-Chu Wong, *et al.* 2000. Incremental Document Clustering for Web Page Classification. *Proceedings of IEEE International Conference on Info. Society in 21st century: emerging technologies and new challenges*. pp. 0-20.