Comparing the Clustering Efficiency of ACO and K-Harmonic Means Techniques

M. Divyavani[1] and T.Amudha[2]

[1]M.Phil Research Scholar, Department of Computer Application, Bharathiar University, Coimbatore, India

divicse07@gmail.com

[2]Assistant Professor, Department of Computer Application, Bharathiar University, Coimbatore, India

amudhaswamynathan@buc.edu.in

Abstract

In the last two decades, many advances on the computer sciences have been based on the observation and emulation of processes of the natural world. The nature inspired methods like ant based clustering techniques have found success in solving clustering problems. They have received special attention from the research community over the recent years because these methods are particularly suitable to perform exploratory data analysis. The clustering is an important technique that has been studied in various fields with many applications such as image processing, marketing, data mining and information retrieval. Recently, the various algorithms inspired by nature are used for clustering. This paper focuses on the behavior of clustering procedures in two approaches, ant based clustering algorithm and K-harmonic means clustering algorithm. The two algorithms were evaluated in two of well- known benchmark data sets. Empirical results clearly show that ant clustering algorithm performs well compared to another technique called K-Harmonic means clustering algorithm.

Keywords

Biological algorithms, data mining and clustering techniques, ant colony optimization clustering algorithm (ACOC), K-Harmonic Means Clustering algorithm (KHM)

1. INTRODUCTION

In bio- inspired artificial intelligence concepts like the swarm intelligence approach, where the behavior of social insects like ants or bees is copied, communication is carried out exclusively through the environment. The ants, bees, termites, and wasps are classified as social insects because they live in colonies. Every individual in a social insect colony seems to act independently of the others, but still the colony functions as an organized unit. These social colonies can be thought of as natural problem solving systems having collective intelligence [4].

The nature inspired methods like ant based clustering techniques have found success in solving clustering problems. They have received special attention from the research community over the recent years because these methods are particularly suitable to perform exploratory data analysis. Since there is a lot of investigation to perform on this field – the research nowadays concentrates on improving performance, stability, convergence, speed robustness and other key features that would allow applying these methods in real world applications. The main research on the nature inspired methods does not focus on the strict modeling of the natural processes; it merely focuses

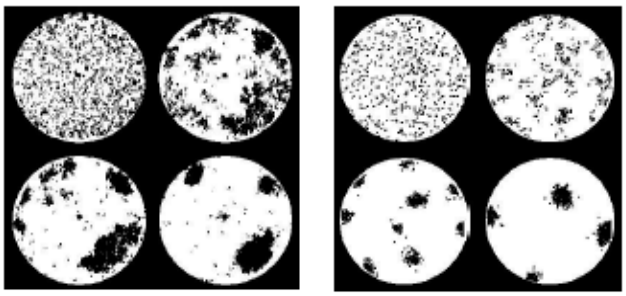
DOI : 10.5121/ijcsea.2011.1409

on using the best ideas to improve the convergence and accuracy of such methods [4]. The study of ant colonies has offered great insight in this aspect.

Data mining, as well as its synonyms knowledge discovery and information extraction is frequently referred in the literature as the process of extracting interesting information or patterns from large data bases. There are two major issues in data mining research and applications; patterns and interest. The techniques of pattern discovery include classification, association, outlier and clustering. Data mining may also be viewed as the process of turning the data into information, the information into action, and the action into value or profit. That is, mining those actionable patterns that the user can act on them to his advantage [21]. The clustering techniques are used to discover natural groups in the data set and identify abstract structures that may reside in these groups. Data clustering is a useful process to extract meaning from sets of unlabeled data or to perform data exploration for pattern recognition [13].

The goal of data clustering is to group objects that are similar to one another and separate those that are not. Unlike the classification task, the set of labels are not known in advance [11]. Figure 1 shows a clustering procedure. The typical cluster analysis consists of four steps with a feedback pathway. These steps are closely related to each other and affect the derived clusters [15]. There exist a large number of clustering algorithms in the literature including K-Means, K-Harmonic Means, K-Medoids, CURE, CACTUS, CHAMELEON, and DBSCAN. No single algorithm is suitable for all types of objects, nor all algorithms appropriate for all problems. The study of ant based clustering algorithm has offered great insight in this clustering aspect [1].

Ant algorithms are a class of the algorithms based on artificial swarm intelligence, which is inspired by the collective behavior of social insects. Different ant algorithms have been developed and applied to a variety of problems. For instance, such approaches were successfully used in real life problems like job scheduling and network routing [20].



Real ants *Messor sancta* build clusters starting from randomly located corpses

Simulated ants build clusters starting from randomly located items

Figure 1 Real ant clusters the bodies of deal ants [12]

The ant clustering algorithms originated from the studies of ant clustering of dead bodies. They were introduced by Deneubourg *et al*., and improved by Handl *et al*., and are mainly applied to solve data clustering problems [20]. The above Figure 1 shows how to ants cluster the data.

According to O.A. Mohamed Jafer and R. Sivakumar (2010), the nature inspired methods like ant based clustering techniques and swarm intelligence have found success in solving clustering problems. [12]. The Urszula Boryczka (2008) , written as among the many bio-inspired techniques, ant clustering have received special attention, especially because they still require much investigation to improve performance, stability and other key features that would make

such algorithms mature tools for data mining. [19]. Salima Ouadfel and Mohamed Batouche (2007), told that ants algorithm dynamically cluster pixels into distinctive independent groups within which similar pixels are closely placed in the same cluster which is gave better clustering quality compared to those obtained from KMeans algorithm [17]. Aranha and Claus de Castro (2006) stated that Ant-inspired techniques have shown greater promise to the clustering problem. In fact, ant-based clustering techniques are competitive with traditional ones [1].

According to Jeffrey W. Seifert (2004) Data mining is the use of sophisticated data analysis tools to discover previously unknown, valid patterns and relationships in large data sets. The data mining consists of more than collecting and managing data, it also includes analysis and prediction and it can be performed on data represented in quantitative, textual, or multimedia forms [17]. The Zengyou He, Xiaofei Xu and Shengchun Deng (2003) stated that Data mining may also be viewed as the process of turning the data into information, the information into action, and the action into value or profit. That is, mining those actionable patterns that the user can act on them to his advantage [21]. Then Amuel Sambasivam and Nick Theodosopoulos (2006) opined that, Data mining involves the use of search engine algorithms looking for hidden predictive information, patterns and correlations within large databases. The technique of data clustering divides datasets into mutually exclusive groups [16].

Periklis Andritsos (2002) defined that Cluster analysis organizes data by abstracting underlying structure either as a grouping of individuals or as a hierarchy of groups. The representation can then be investigated to see if the data was grouped according to preconceived ideas or to suggest new experiments [12]. The Manying Qiu (2004) told that, Clustering is different from classification because clustering does not require predefined classes. The records are grouped based on self-similarity. It is up to the user to interpret the resulting clusters. Clustering is undirected knowledge discovery—no target variable is defined [9] and according to V. Estivill-Castro (2004), K-means clustering algorithm has been adopted as the prototype of iterative model based clustering because of its speed, simplicity and capability to work within the format of very large database [4].

B. Gillner (2007) stated that, the ant colony algorithm observed in the wild has been attributed with the remarkable habit of accumulating larvae and food in a distinctive order, reminiscent of clustering's of sets of data [6]. Y. Kao · S.C. Fu (2005) stated, that the ant-based clustering in order to resolve machine cell formation problems. The three-phase algorithm mainly utilizes distributed agents which mimic the way real ants collect similar objects to form meaningful piles. [8]. Zahra Sadeghi, Mohammad Teshnehlab and Mir Mohsen Pedram (2007) defined K-ants clustering algorithm which used clustering with ants in which the number of clusters must be provided for it in advance. The clustering was done using a square grid. Each ant has a load list that must be filled with the members of one cluster. So every ant is supposed to search for one distinct cluster [21].

This paper is organized as follows: The section II presents the objective and methodology by which ant based clustering technique, k-harmonic means clustering technique and the bench mark instances taken for clustering were applied in this research work. The implementation results of ACOC algorithm and the KHM algorithm were presented analyzed and interpreted in section III The last section IV concluded the whole paper and pointed out the major strength of this work, contribution to the domain knowledge and direction for future research.

2. PROBLEM FORMULATION AND METHODOLOGY

The main objective of this research work is to implement two clustering algorithms, one bio-inspired clustering technique and one traditional clustering technique and to study their clustering competency.

2.1 ACOC Algorithm

In the ACOC algorithm, an artificial ant colony simulates the pheromone trail following the behavior of real ants. Artificial ants move on a synthetic map representing a specific problem to construct solutions successively [20].

In the ACOC algorithm, the solution space is modeled as a graph of object-cluster node matrix. The number of rows equals *m*, and the number of columns equals *g*. Each node denoted by N (*i*, *j*) meant that data object *I* would be assigned to cluster *j*. Artificial ants can stay at only one of *g* nodes for each object. Figure 3 illustrates an example of construction graphs for clustering problems, where hollow circles denote unvisited nodes and solid circles represent visited nodes. A string is used to represent solutions built by ants. Considering the clustering result of Fig- 4, the corresponding solution string is (2, 1, 2, 1, 3, and 3).

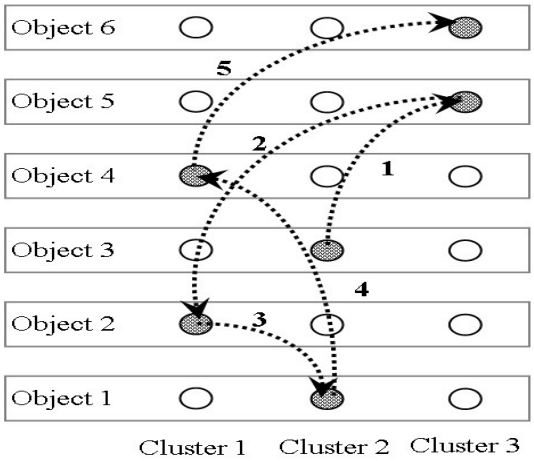


Figure 2 Construction graph for ACOC [20]

On the graph, each ant moves from one node to other, deposits pheromone on nodes, and constructs a solution in a stepwise way. At each step, an ant randomly selects an ungrouped object and adds a new node to its partial solution by considering both pheromone intensity and heuristic information. The memory list (tb^k) can prevent a data object from being clustered more than once by an ant. When the memory list is full, it means that the ant has complete solution construction. The moving sequence of the example in Figure 2 is marked by the numbers next to the dotted arcs.

2.1.1 Steps in ACOC Algorithm

1. *Initialize the pheromone matrix: The elements of the pheromone matrix (PM) are set to arbitrarily chosen small values (τ_0).*
2. *Initialize all ants: Start a new iteration. Reset the memory list (tb^k), cluster center matrix (C^k) and weight matrix (W^k) for each ant, where k=1˜R. R is the total number of ants, R≤m.*
3. *Select data object i: Each ant randomly selects a data object, I, that is not in its memory list.*
4. *Select cluster j: To determine j for a selected I, two strategies, exploitation and exploration, can be applied. The first ants to move in a greedy manner to a node whose product of pheromone level and heuristic value is the highest (equation (7)). The latter is to allot probabilities to candidate nodes, and then let an ant choose one*

of them in a stochastic manner according to equation (8). The more promising a node, the higher probability it has. Ants choose one of these strategies by using equation (7) with a priori defined probability q_0 and a randomly generated probability q. based on equations (7) and (8), ants can determine the value of j,

Note that $\eta^k_{i,j}$= 1/d^k (i, j) and is the heuristic value of N(i,j) for ant k. The distance between object i and center j of ant k, d^k (i, j), is defined in equation (9).

$$j = \begin{cases} \arg \max_{u \in N_i} \left\{ [\tau(i,u)][\eta^k(i,u)]^\beta \right\} & \text{if } q \leq q_0 \\ S & \text{otherwise} \end{cases}$$

Where Ni is the set of g nodes belonging to data object I, and the value of S is chosen according to equation (8).

$$P^k(i,j) = \frac{[\tau(i,u)]\,[\eta^k(i,u)]^\beta}{\sum_{j=1}^{g} [\tau(i,u)]\,[\eta^k(i,u)]^\beta}$$

Where β is the parameter specifying the relative weight of $\eta^k_{ij},\ \beta > 0.$

$$d^k(i,j) = \sqrt{\sum_{v=1}^{n} \left(x_{iv} - c^k_{jv}\right)^2}$$

Where c^k_{jv} *refers to the value of attribute v of cluster center j of ant k.*

5. *Update ants' information: Update the memory list (tb^k), weight matrix (W^k, use equation (5)) and cluster matrix (C^k, use equation (6)) of each ant.*
6. *Check memory list of each ant: Check if the memory list of each ant is full. If it is not, then go back to step3; otherwise, go to step7.*
7. *Calculating objective function values: Calculate the objective function value of each ant, J^k values. The best solution is called iteration-best solution. It is compared with the best so far solution, and the better one will be the new best so far solution.*
8. *Update pheromone trails: Update the pheromone matrix, PM. The global updating rule is applied, and only the elitist ants are allowed to add pheromone at the end of each iteration. The trail updating equation is defined below.*
   Where ρ is the pheromone evaporation rate, (0.0 < ρ <1.0), t is the iteration number, r is the number of elitist ants, and $\Delta\tau_{ij}^h = i/J^h$.
9. *Check termination condition: If the number of iterations exceeds the maximum iteration number, then stop and output the best so far solution, otherwise go to step2.*

2.2 KHM Algorithm

K-harmonic means (KHM) is a more recent algorithm presented by Zhang in 2000. This algorithm minimizes the harmonic average from all points in N to all centers in K [3], which uses the Harmonic Averages of the distances from each data point to the centers as components to its performance function.

Given a set N of n data points in d dimensional space, it should be determined how to assign a set K of k points, called centers, in N so as to optimize based on some criterion. In most cases, it is natural to assume that N is much greater than K and d is relatively small. This formulation is an example of unsupervised learning. The system will create grouping based only on the criterion and the information contained in the n data point. In this algorithm to describe the class of KHM with parameter *p* that is power associated with the distance calculation. In the standard KM algorithm p would be 2 because the distance calculation is given by *squared* distance $||x_i - c_j||^2$. It was found that KHM works better with values of $p > 2$. The harmonic average is defined as

HA ({a_1... aK}) = K / [SUM over k = 1 to K (1 / a_k)] (1)

This function has the property that if any one element in $a_1..a_K$ is small, the Harmonic Average will also be small. If there are no small values the harmonic average will be large. It behaves like a minimum function but also gives some weight to all the other values. The objective function of KHM is given by:

Minimize [SUM over i = 0 to N [HA ($||x_i - c_j||^2$ for all c_j in K)]] (2)

Where HA () is the harmonic average for each data point. Unlike KM, this algorithm uses information from all of the centers in K to calculate the harmonic average for each point in N. This means that no center completely owns a point, but rather partially influences the harmonic average for each point.

2.2.1 Complete Pseudo Code for KHM

Data Structures:

N: n by d+1 array - contains static information about data set

Nmin: n element array which holds the minimum distance to any center

K: k by d array that holds information about centers

M: n by k array that holds distance from all point in N to all points in K

Temporary Arrays (Could be reduced but shown for simplicity)

U: n element array

Q: n by k temporary array

QQ: k element array

R: n by k temporary array

RR: k element array

T: n by k

p: KHM parameter

Initialization

1. *Create an initial K:*
   Choose any k points from N

Main Loop

2. *Fill Matrix M:*
   Calculate distances from all points in N to all centers in K

3. *Compute Nmin:*
   Find minimum distance for to any center for each point in N

4. *Recompute Harmonic Averages and Update K:*

```
For each point (j = 0 to n)
    For each center (i = 0 to k)
        U[j] = U[j] + (Nmin[j]/N[j,I])
U[j] = U[j] – 1;
For each center (i = 0 to k)
    For each point (j = 0 to n)
        Q[j,I] = [(Nmin[i]^(p-2) * (Nmin[i]/N[j,i])^(p+2)] /  [(1 + U[j]^p)^2]
For each center (i = 0 to k)
    For each point (j = 0 to n)
        QQ[i] += Q[j,I]
For each center (i = 0 to k)
    For each point (j = 0 to n)
        R[j,I] = Q[j,i] / QQ[i]
For each center (i = 0 to k)
    For each point (j = 0 to n)
        K[i] = K[i] + R[j,i]*N[j]
```

5. *If no center is updated in step 4 then stop, as the algorithm has converged*

The algorithm above is given in a simplified form as to show all the temporary values that are calculated by taking partial derivative for the Harmonic Average Function. Each nested loop in Step 4 represents the one of the five decomposed equations from Equation 6 [3].

3. IMPLEMENTATION RESULTS AND DISCUSSION

The experimental results and comparative study of the two algorithms are presented in this section. The performance of the two algorithms was evaluated by testing on two datasets, NURSERY and SOLAR dataset. These datasets were selected from the website of UCI repository of machine learning databases.

These algorithms have been coded in java platform, java extended support a good for clustering the data objects while executing the program. It is worth to use java for clustering techniques. The SQL Server 2000 was used in this research work to construct the two database namely NURSERY database and SOLAR database. These databases contain the different types of instance values, and attribute of the two categorical datasets. The clustering results of the two algorithms on the test sets are compared using two evaluation measures called Entropy and F-Measure.

3.1 Entropy

Entropy is used to measure the quality of the clusters. Let CS be a clustering solution. For each cluster, the class distribution of the data is calculated first, i.e., for cluster j is computed. The "probability" p_{ij}, that denotes whether a member of cluster j belongs to class i is computed. Then using this class distribution, the entropy of each cluster j is calculated using the standard formula

$$E_j = -\sum_i \rho_{ij} \log(\rho_{ij}) \quad (3)$$

where the sum is taken over all classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster:

$$E_{CS} = \sum_{J=1}^{m} \frac{n_J * E_j}{n} \quad (4)$$

where n_j is the size of cluster j, m is the number of clusters, and n is the total number of data points [2].

3.2 F measure

The second external quality measure is the F measure, a measure that combines the precision and recall ideas from information retrieval. Each cluster can be treated as if it were the result of a query and each class as if it were the desired set of data items for a query. Next, recall and precision of that cluster is calculated for each given class. More specifically, for cluster j and class i

$$\text{Recall } (i, j) = n_{ij} / n_i \quad (5)$$
$$\text{Precision } (i, j) = n_{ij} / n_j \quad (6)$$

Where n_{ij} is the numbers of members of class i in cluster j, n_j is the number of members of cluster j and n_i is the number of members of class i. The F measure of cluster j and class i is then given by

$$F(i, j) = (2 * \text{Recall}(i, j) * \text{Precision}(i, j)) / ((\text{Precision}(i, j) + \text{Recall}(i, j)) \quad (7)$$

For an entire hierarchical clustering the F measure of any class is the maximum value it attains at any node in the tree and an overall value for the F measure are computed by taking the weighted average of all values for the F measure as given by the following.

$$F = \sum_n n_i \max \{F(I,j)\} \quad (8)$$

where the maximum is taken over all clusters at all levels, and n is the number of data items [2].

For the two algorithms, the experimental results were shown in the following tables. The study compared the performance of the KHM and ACOC. For each test problem, these two algorithms were performed 12 times (distinct runs) individually. The parameter values used in KHM and ACOC were: nof / R =2 or 3 or 4 or 5 (declared number of clusters), iter = 25, minimum iteration = 5, maximum = 25. The 12 distinct runs were grouped in Table 1

Table 1: 12 Distinct Runs

Cluster Groups	Number of Clusters	Iteration Name
A	2	5
		15
		25
B	3	5
		15
		25
C	4	5
		15
		25
D	5	5
		15
		25

The test sets were tested by four cluster groups namely A, B, C, D. The first group A had two clusters with three different iterations 5, 15, 25. The second group B had three clusters with three iterations 5, 15, 25. The third group C had four clusters with three iterations 5, 15, and 25. Finally, D group had five clusters with three iterations 5, 15, 25 was evaluated. The four clusters were tested with a maximum of 25 iterations by implementing ACOC algorithm and KHM algorithm. The clustering efficiency of the both ACOC and KHM algorithm was evaluated by testing on two datasets. For the real life datasets, instances were selected from the UCI repository of machine learning databases namely NURSERY and SOLAR dataset. The first dataset is composed of Multivariate and it has 12960 instance values, while the second also Multivariate dataset which has 1066 instances.

3.3 Nursery Dataset

Table 2 shows the computational results of NURSERY data set in ACOC and KHM and the Table 3 shows the standard deviation of the performance measures in ACOC and KHM for NURSERY data set along with graph is presented Figure 3 and Figure 4

Table 2: Computational results of NURSERY dataset

Cluster Groups	No.of Clusters	Iteration Name	Objective function values				Objective function values			
			ACOC				KHM			
			Entropy	Precision	Recall	F-Measure	Entropy	Precision	Recall	F-Measure
A	2	5	0.159	0.582	0.582	0.707	0.174	0.371	0.371	0.453
		15	0.173	0.102	0.102	0.124	0.174	0.371	0.371	0.453
		25	0.162	0.557	0.557	0.675	0.316	0.135	0.135	0.165
B	3	5	0.105	0.104	0.073	0.104	0.094	0.525	0.381	0.532
		15	0.183	0.181	0.127	0.180	0.163	0.857	0.660	0.923
		25	0.162	0.180	0.127	0.180	0.166	0.517	0.386	0.535
C	4	5	0.078	0.468	0.275	0.410	0.202	0.242	0.184	0.225
		15	0.135	0.811	0.477	0.711	0.122	0.176	0.101	0.152
		25	0.174	0.904	0.615	0.918	0.451	0.543	0.411	0.504
D	5	5	0.133	0.764	0.520	0.615	0.113	0.807	0.400	0.621
		15	0.141	0.656	0.346	0.433	0.142	0.432	0.227	0.283
		25	0.113	0.255	0.125	0.195	1.198	0.255	0.151	0.195

Table 3: Standard deviation of performance measures

Quality measures	Objective function values ACOC		Objective function values KHM	
	Avg	Stdev	Avg	Stdev
Entropy	0.130	0.052	0.646	0.551
F-Measure	0.511	0.406	0.537	0.384

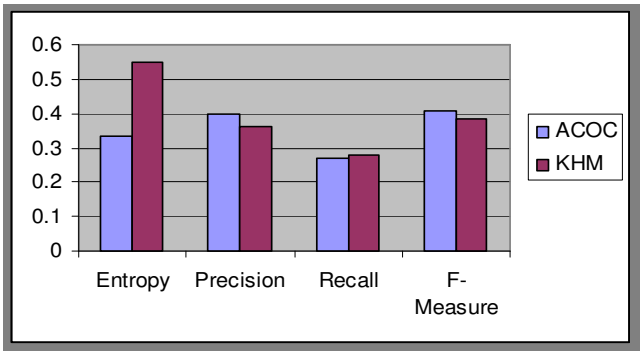


Figure 3 Comparsion graph of Standard deviation in ACOC and KHM

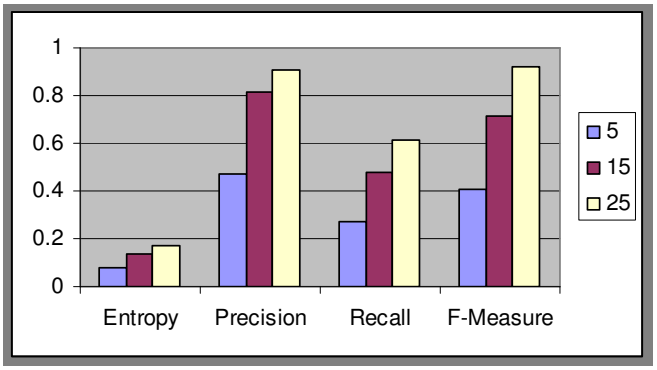


Figure 4 Best cluster group in ACOC (C-Group)

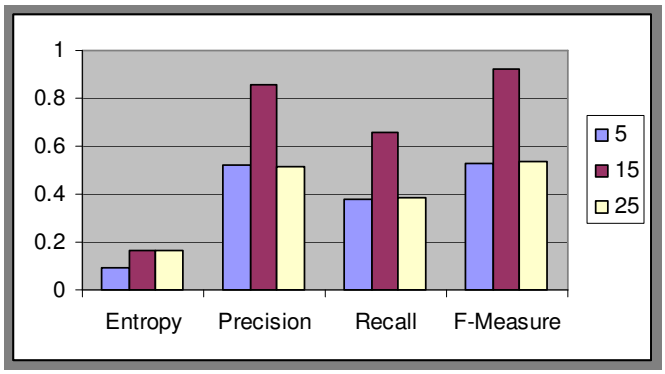


Figure 5 Best cluster group in KHM (B-Group)

It is clearly understandable from Table 2 and Table 3 that the ACOC gave comparable solutions. It was found that ACOC has a consistent performance over all the iterations with respect to the quality measures. The ACOC illustrated the best standard deviations for entropy with the value of 0.052 whereas the best standard deviation for F-measure was 0.384 as given by KHM. Table 3 shows the standard deviation for corresponding computation time with graph and Figure 4, Figure 5 shows the best cluster groups in ACOC and KHM algorithm.

Table 4: Standard deviation for computation time

Cluster groups	CPU time Values		CPU time Values	
	ACOC		KHM	
	Average	Stdev	Average	Stdev
A	72.394	0.888	67.102	4.824
B	71.569	0.184	70.356	1.384
C	71.915	0.140	71.918	0.112
D	71.696	0.263	62.862	8.991

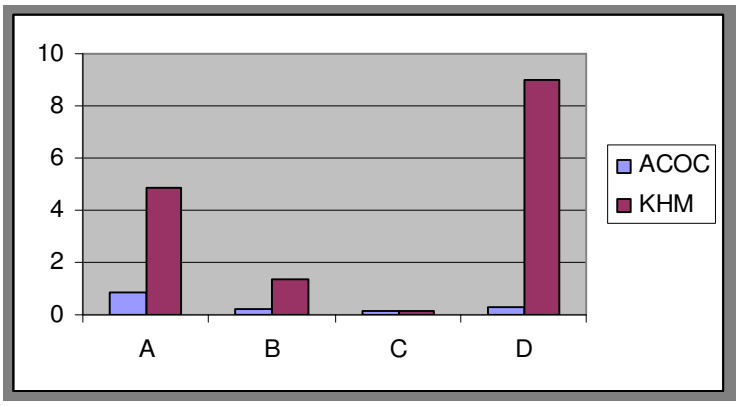


Figure 6 Comparsion graph of standard deviation for computation time

It is clearly understandable from Table 4 that the ACOC algorithm has performed consistently at all level of iterations. The ACOC located the best time as 71.382 to form three clusters in twenty five iterations. The KHM found the best time as 53.870 to form five clusters in five iterations. The execution time for ACOC is relatively higher than KHM. Even though ACOC has taken larger processing time, its standard deviation was found to be much better than KHM. In ACOC, the highest standard deviation of computation time was found as 0.888 whereas is KHM, it was 8.991.

3.4 Solar Dataset

Table 5 the computational results of SOLAR data set in ACOC and KHM and the Table 6 shows the standard deviation of the performance measures in ACOC and KHM for NURSERY data set along with graph is presented Figure 6 and Figure 7.

Table 5: Computational results of SOLAR dataset

Cluster Groups	No.of Clusters	Iteration Name	Objective function values				Objective function values			
			ACOC				KHM			
			Entropy	Precision	Recall	F-Measure	Entropy	Precision	Recall	F-Measure
A	2	5	0.321	0.136	0.136	0.136	0.312	0.138	0.138	0.138
		15	0.292	0.141	0.141	0.173	0.300	0.141	0.141	0.141
		25	0.321	0.136	0.136	0.136	0.265	0.139	0.140	0.139
B	3	5	0.164	0.244	0.172	0.199	0.167	0.147	0.209	0.170
		15	0.285	0.424	0.298	0.344	0.289	0.362	0.256	0.295
		25	0.364	0.223	0.157	0.222	0.270	0.283	0.201	0.231
C	4	5	0.115	0.256	0.103	0.126	0.125	0.404	0.231	0.284
		15	0.200	0.310	0.179	0.219	0.217	0.700	0.401	0.492
		25	0.258	0.323	0.231	0.283	0.280	0.903	0.655	0.759
D	5	5	0.173	0.541	0.271	0.420	0.123	0.375	0.188	0.237
		15	0.255	0.607	0.304	0.384	0.392	0.650	0.325	0.412
		25	0.175	0.740	0.226	0.286	0.175	0.384	0.192	0.243

Table 6: Standard deviation of performance measures

Quality measures	Objective function values		Objective function values	
	ACOC		KHM	
	Average	Stdev	Average	Stdev
Entropy	0.203	0.100	0.217	0.140
F-Measure	0.273	0.147	0.448	0.310

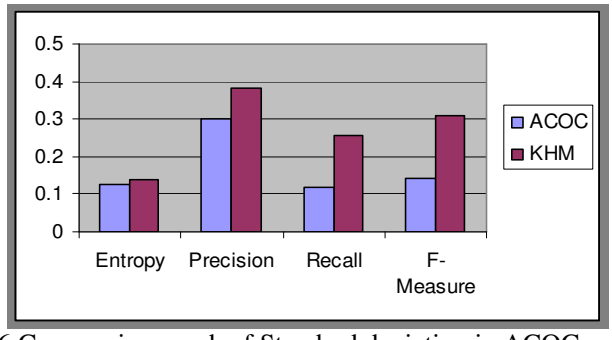


Figure 6 Comparsion graph of Standard deviation in ACOC and KHM

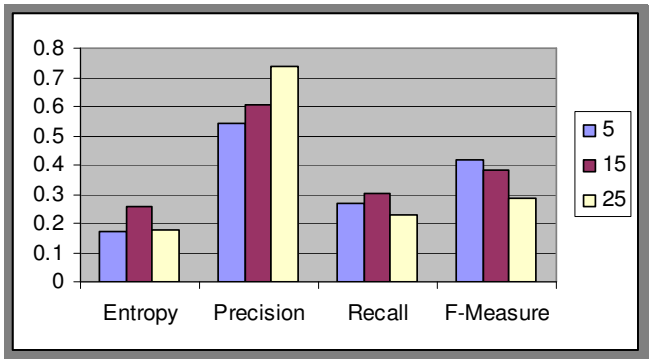


Figure 7 Best cluster group in ACOC (D-Group)

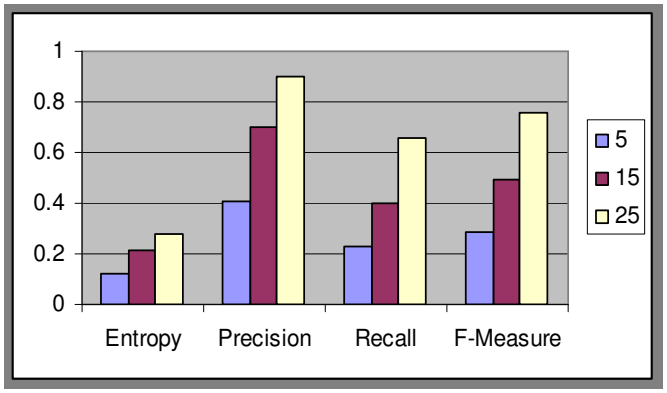


Figure 8 Best cluster group in KHM (C-Group)

It is clearly understandable from Table 5 and Table 6 that the ACOC gave comparable solutions. It was found that ACOC has a consistent performance over all the iterations with respect to the quality measures. The ACOC illustrated the best standard deviations for entropy with the value of 0.100 and also given the best standard deviation for F-measure 0.147 with compared to KHM. Table 5 shows the standard deviation for corresponding computation time with graph Figure 7, Figure 8 shows the best cluster groups in ACOC and KHM algorithm.

.Table 7 Standard deviation for computation time

Cluster groups	CPU time Values		CPU time Values	
	ACOC		KHM	
	Average	Stdev	Average	Stdev
A	46.125	0.076	66.617	15.027
B	50.312	0.025	81.344	0.104
C	81.258	0.122	46.652	0.241
D	81.748	0.017	68.938	13.787

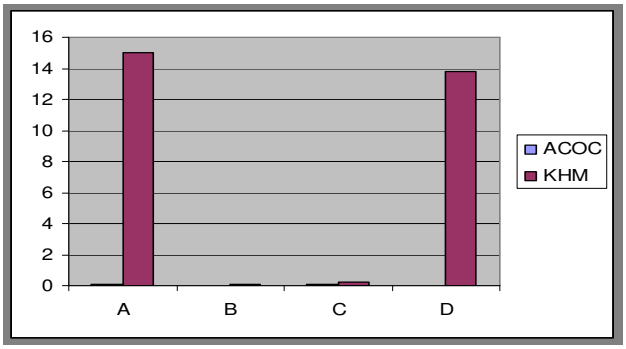


Figure 8 Comparsion graph of standard deviation for computation time

It is clearly understandable from Table 7 that the ACOC algorithm has performed consistently at all level of iterations. The ACOC located the best time as 46.049 to form two clusters in five iterations. The KHM found the best time as 46.411to form four clusters in twenty five iterations. The execution time for ACOC is relatively higher than KHM. Even though ACOC has taken larger processing time, its standard deviation was found to be much better than KHM. In ACOC, the highest standard deviation of computation time was found as 0.122 whereas is KHM, it was 15.027.

5. CONCLUSION AND FUTURE DIRECTIONS

In NURSERY dataset, the ACOC has a consistent performance over all the iterations with respect to the quality measures and it also illustrated the best standard deviations for entropy. KHM has shown the best standard deviations for F-Measure in this dataset. The execution time for ACOC is relatively higher than KHM. Even though ACOC has taken larger processing time, its standard deviation for processing time was found to be much better than KHM. In SOLAR dataset, the ACOC has outperformed KHM over all the iterations with respect to the quality measures as well as the best standard deviations for entropy and F-Measure. The execution time for ACOC is relatively higher than KHM but its standard deviation for execution time was found to be much better than KHM. The ACOC has larger processing time compared to KHM and this problem need to be solved in the future. The standard deviation of KHM is larger than the ACOC. This issue is to be addressed and improved in future.

6. REFERENCES

[1] Aranha and Claus de Castro, "*A Survey on using Ant-Based techniques for Clustering*", 2006.

[2] Banumathy D., and Nithyakalyani S., *"An Improved K-Mean Clustering Hybrid with Ant-Colony Optimization",* KSR College of Engineering, Tamil Nadu, 1st National conference on Intelligent Electrical Systems (NCIES'09) 24-25 April 2009.

[3] Douglas Turnbull, *"K-Means & K-Harmonic Means: A Comparison of Two Unsupervised Clustering Algorithms*", CSE 202 – Project, 2002.

[4] Elisa Valentina Onet and Ecaterina Vladu, "Nature Inspired Algorithms and Artificial Intelligence", Journal of Computer Science, 2005.

[5] Estivill-Castro V., and Yang J., "*Fast and Robust General Purpose Clustering Algorithms",* Griffith University, Nathan, QLD 4111, Australia. University of Western Sydney, Campbelltown NSW 2560, Australia, Data Mining and Knowledge Discovery, 8,127-150, 2004, Kluwer Academic Publishers. Manufactured in the Netherlands.

[6] Gillner B., *"A Comparative Study Of Ant Clustering Algorithms",* October 22, 2007.

[7] Jeffrey W. Seifert, *"Data Mining: An Overview",* Analyst in Information Science and Technology Policy, Resources, Science, and Industry Division, CRS Report for Congress, 2004.

[8] Kao Y., Fu S.C., **"***An Ant-Based Clustering Algorithm for Manufacturing Cell Design's,* Published online: 21 December 2005 © Springer-Verlag London Limited 2005.

[9] Manying Qiu, Steve Davis and Fidelis Ikem, *"Evaluation of Clustering Techniques in Data Mining Tools"*, Virginia State University, Clemson University, Volume V, No 1, 2004.

[10] Manying Qiu, Steve Davis and Fidelis Ikem, *"Evaluation of Clustering Techniques in Data Mining Tools"*, Virginia State University, Clemson University, Volume V, No 1, 2004.

[11] Marcio Frayze David and Leandro Nunes de Castro, "*A New Clustering Boids Algorithm for Data Mining"*, Mackenzie University, Brazil 2008.

[12] Mohamed Jafer O.A., and Sivakumar R., "*Ant-Based Clustering Algorithms: A Brief Survey***"**, International Journal of Computer Theory and Engineering, Vol 2, No.5, October, 2010, 1793-8201.

[13] Periklis Andritsos, *"Data Clustering Techniques",* University of Toronto, Department of Computer Science, 2002.

[14] Pro Dromidis A.L., Chan P.K., and Stolfo S.J., *"Meta-Learning in Distributed Data Mining Systems: Issues and Approaches",* Advances in Distributed and Parallel Knowledge Discovery, AAAI/MITPress 2000.

[15] Rui Xu, *"Survey of Clustering Algorithms***",** IEEE and Donald Wunch II, Fellow, IEEE, IEEE Transactions on Neural Networks, vol. 16, No. 3, May 2005.

[16] Salima Ouadfel and Mohamed Batouche, *"An Efficient Ant Algorithm for Swarm-Based Image Clustering",* University of Batna, Algeria, Computer Vision Group, LIRE Laboratory, University of Constantine, Algeria (2007), Journal of Computer Science 3 (3): 162-167, 2007 ISSN 1549-3636 © 2007 Science Publications.

[17] Samuel Sambasivam and Nick Theodosopoulos, *"Advanced Data Clustering Methods of mining Web Documents", A*zusa Pacific University, Azusa, CA, USA, and London, UK, Issues in Informing Science and Information Technology volume 3, 2006.

[18] Sudipto Guha, Rajeev Rastogi and Kyuseok Shim, *"ROCK: A Robust Clustering Algorithm for Categorical Attributes", S*tandford University Standford, Bell Laboratories, Murray, 1997.

[19] Urszula Boryczka, *"Ant Clustering Algorithm",* Institute of Computer Science, University of Silesia, Sosnowiec, Poland, Intelligent Information Systems 2008, ISBN 978-83-60434-44-4, pages 377-386.

[20] Yucheng Kao and Kevin Cheng, "*An ACO-Based Clustering Algorithm***",** Tatung University, Taipei, Taiwan (2006).

[21] Zahra Sadeghi and Mohammad Teshnehlab, Mir Mohsen Pedram, "*K-Ants Clustering- A New Strategy Based on Ant Clustering",* Islamic Azad University 2007.

[22] Zengyou He, Xiaofei Xu and Shengchun Deng, "*Data Mining for Actionable Knowledge: A Survey",* Harbin Institute of Technology, China, 2003.

Authors

Ms. M. Divyavani received her B.Sc degree in Computer Science, M.Sc degree in Computer Science, M.Phil in Computer Science in 2007, 2009, and 2011respectively, from Bharathiar University, Coimbatore, India. She has attended National conferences. Her area of interest includes OOPS concept, Agent based computing and Bio-inspired computing.

Mrs. T. **Amudha** received her B.Sc Degree in Physics, Masters Degree (MCA) in Computer Applications and M.Phil in Computer Science in 1995, 1999, and 2003 respectively, from Bharathidasan University, India. She has qualified UGC-NET for Lectureship in 2003 and is currently pursuing her doctoral research at Bharathiar University in the area of Biologically inspired computing. She is currently working as Asst. Professor in the Department of Computer Applications, Bharathiar University, Coimbatore and has 12 years of academic experience. She has more than 20 research publications for her credit in International/ National Journals & Conferences. Her area of interest includes Software Agents, Bio-inspired computing and Grid computing.