Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

An Adaptive Ant-Based Clustering Algorithm with Improved Environment Perception

I. El-Feghi*, M. Errateeb**, M. Ahmadi*** and M. A. Sid-Ahmed***

idrise@ee.edu.ly, ratibmoh@ptqi.edu.ly, ahmadi@uwindsor.ca, Ahmed@uwindsor.ca
*EE. Dept, Al-Fateh University, Tripoli- Libya, ** Petroleum Training and Qualifying Institute, Tripoli-Libya, *** ECE Department, University of Windsor, Windsor, On., Canada

***Abstract*— Data clustering plays an important role in many disciplines, including data mining, machine learning, bioinformatics, pattern recognition, and other fields. When there is a need to learn the inherent grouping structure of data in an unsupervised manner, ant-based clustering stand out as the most widely used group of swarm-based clustering algorithms. Under this perspective, this paper presents a new Adaptive Ant-based Clustering Algorithm (AACA) for clustering data sets. The algorithm takes into account the properties of aggregation pheromone and perception of the environment together with other modifications to the standard parameters that improves its convergence. The performance of AACA is studied and compared to other methods using various patterns and data sets. It is also compared to standard clustering using a set of analytical evaluation functions and a range of synthetic and real data collection. Experimental results have shown that the proposed modifications improve the performance of ant-colony clustering algorithm in term of quality and run time.**

Keywords*— Adaptive Ant Colony, Swarm Intelligence, Clustering, Optimization.

I. INTRODUCTION

Clustering is defined as dividing a set of data points into non-overlapping groups, or clusters, of points, where points in a cluster are “more similar” to one another than to points in other clusters. The term “more similar,” when applied to clustered points, usually means closer by some measure of proximity. When a dataset is clustered, every point is assigned to some cluster, and every cluster can be characterized by a single reference point, usually the average of all points in the cluster. Any particular division of all points in a dataset into clusters is called a partitioning. Data clustering has gone through vigorous development in the last few years.

Clustering is an essential tool in data mining and knowledge discovery. used in a wide range of applications, such as marketing, biology, psychology, astronomy, and image processing. For example, in biology it is used to form taxonomy of species based on their features and to group the set of co-expressed genes together into one group [2]. In image processing it is used to segment texture in images to differentiate between various regions or objects [3]. Other applications include speech[4] and character recognition [5]. In business, clustering is used as a data mining and knowledge discovery tool for knowledge for making credit decisions and clustering browsers into different groups[6].Clustering is also widely used for data compression in image processing, which is also known as vector quantization [7]. Economic forecasting and time series clustering provides still another venue in data modeling [8].

Although there are many clustering methods already available, clustering algorithms can be broadly divided into two main categories; namely: *hierarchical* and *partitioning*. Hierarchical algorithms produce a hierarchy of clusters by finding successive clusters using previously established clusters. Hierarchical algorithms can be further divided into agglomerative "bottom-up" and divisive "top-down". Agglomerative algorithms start with each element as a separate cluster and merge elements into successively larger clusters. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters [9]. The main advantage of hierarchical algorithms is that they are more versatile as they do not require an a priori definition of the number of clusters [10].

On the other hand, partitioning algorithms find clusters by minimizing an objective function model defined as the sum of the distances between the data points and the centers of the clusters that they belong to. Portioning algorithms include crisp clustering and fuzzy clustering. In crisp clustering, such as *k-means*, data vectors can belong to only on cluster while in fuzzy clustering, *c-means*, data vectors can belong to more than one cluster with a degree of belonging or membership. Although crisp clustering is very popular because of its speed, fuzzy approach is both less prone to local minima and more informative as a technique for data clustering [11]. The disadvantage of portioning clustering is the number of clusters has to be known beforehand. This imposes a challenge to data clustering; it is difficult to know beforehand what data number should be.

Although clustering has been subject to research for many years, the problem is still a challenging and no optimal algorithm that is garneted to work for any data set is yet available. Clustering problem is highly application-dependent; the performance of one algorithm can become meaningful only with the context of a specified problem domain. Hence, it is only through detailed analysis and comparison of algorithms on different kinds of test problems, that we can gain an increased insight into the performance of one particular method.

To help resolve the above mentioned problems, Ant Colony Optimization (ACO) has received much attention from

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

many scholars and been widely applied in different fields since it was proposed by [12]. ACO is an evolutionary computation technique that tends to mimic the behaviour of ants in their search for the shortest paths to food sources. Ants look for optimal solutions by utilizing distributed computing, local heuristics, and knowledge from past experience [13].

In this paper ant colony optimization algorithm is applied to clustering analysis and a novel clustering based on an improved particle swarm optimization algorithm is proposed. Theoretical analysis and experiments show the proposed method Introduced modifications improve the performance in terms of quality and runtime.

This paper is organized as follows: Section 2 describes the basic mathematical model for ACO and how it is used to solve clustering problems. While Section 3 reports the modification to ant-based clustering. Section 4 presents comparison of ant-based clustering to alternative methods. Finally, conclusions of the current work are reported in Section 5.

II. BASIC CLUSTERING MODEL

In ant colony, each ant is to behave individually, moving randomly in space while picking up or dropping corpses. The decision to pick up or drop a corpse is based on local information of the ant's current position. This simple behavior of individual ants results in the emergence of a more complex behavior of clusters formation [14,15]. While these behaviors are still not fully understood, a number of studies have resulted in mathematical models to simulate the clustering behaviors. Based on these simulations, algorithms have been implemented to solve different clustering problems as ants will compile food into one pile and other waste, such as dead ants, into another pile. These different piles will simulate the different clusters. Dorigo, M. and Deneubourg[14] added memorizing function to artificial ants, which enabled ants to keeps track of the types and quantities of objects that an ant has recently encountered.

The first implementations which simulate cemetery formation and brood sorting of ants as mentioned above were inspired by the studies of Chrétien [15] of the ants Lasius. On the basis of Chrétien's observation, the first monetization of this ants' behavior has been done by Deneubourg *et al.*[14], who showed that a simple mechanism involving the modulation of the probability of dropping corpses as a function of the local density of corpses was sufficient to generate the observed sequence of the clustering of corpses The experiments reported where limited to the clustering of one or two types of data items.

The probability P_p for a randomly moving, agent (representing an ant in the model) to pick up an item is given by :

$$P_{pick}(i) = \left(\frac{k_p}{k_p + f(i)} \right)^2 \qquad (1)$$

Where f is the fraction of items the ant perceives in its neighborhood, and $k_p > 0$. When there are only few items in the ant's neighborhood, that is $f << k_p$, then P_p is close to 1; (i.e., the probability of picking-up an item is high when there are not many items in the neighborhood). On the other hand, if the ant observes many objects, that is $f >> k_p$, P_{pick} is close to 0 (i.e., items are unlikely to be removed from dense clusters), and the probability that the ant will pick up an object is small. Each loaded ant has a probability of dropping the carried object, given by:

$$P_{drop}(i) = \left(\frac{f(i)}{k_d + f(i)} \right)^2 \qquad (2)$$

where k_d is another threshold constant, $k_d > 0$. If a large number of items is observed in the neighborhood, i.e. $f >> k_d$, then P_{drop} is close to 1, and the probability of dropping an item is high. If f << k_d then P_{drop} is close to 0.

The fraction of item, f, is calculated by making use of a short term memory for each ant. Each ant keeps track of the last T time steps, and f is simply the number of items observed during these T time step divided by the largest number of items that can be observed during the last item steps. If only one item can be observed during each time step, $f=N_f/T$, where N_f is the number of encountered items.

Deneubourg's work was followed by Lumer and Faieta's model [16] which incorporates a distance or similarity measure between different objects. The model was created to apply the basic model to real-valued vectors, in order to perform exploratory data analysis on databases. The first problem to solve, is to determine an equation representing the distance or dissimilarity, $d(y_a, y_b)$, between a specific object and all other objects within a given radius by using any applicable norm.

In this model, data vectors are scattered randomly on a two-dimensional grid, where a population of ant-like agents randomly move on the grid, while ,observing the surrounding area of s^2 sites, referred to as the s-patch. A number of slight modifications have been introduced that improve the quality of the clustering and, in particular, the spatial separation between clusters on the grid.

Recently Handl and Meyer [20] employed ant-based clustering as the core of a visual document retrieval system for Worldwide web searches in which the basic goal is to classify online documents by contents' similarity. The authors adopted an idea of short-term memory and employed ants with different speeds and allowed ants to jump. In addition, they introduced an adaptive scaling strategy, as well as some further modifications to achieve reliable results and to improve efficiency. Handl and Meyer, have expanded upon Lumer and Faieta's algorithm by introducing: adaptive scaling, ant jumps, stagnation control and `eager ants'.

III. MODIFIED ANT-BASED CLUSTERING ALGORITHM

Starting with a generic algorithm based on earlier work of ant-based clustering, we introduce a number of modifications that overcome the problems identified above. The algorithm that we finally arrive at is operationally different from previous ant-based clustering in a number of key areas. To differentiate the proposed approach from these previously mentioned, we call our algorithm, AACA (Adaptive Ant-Clustering Algorithm), which is more robust in terms of the number of clusters found and tends to converge into good solutions while the clustering process evolved

A. Improve picking and dropping probability functions

The ant colony optimization heuristic simulates the ability of real ants to drop and follow pheromone trails. The presence of pheromones enables the concept of stigmergy[18], a form of indirect communication used by social insects to coordinate their activities, usually by changing the environment with interpretable cues. The key in organizing the colony level behavior is communication via the environment, i.e. via pheromones. The communication is always local on the part of the ants (they always leave the pheromone on their actual location), but it is channeled by the physical environment (diffusion and evaporation). Thus, the individual colony itself has no global communication methods. The colony must therefore achieve its macro-goals by coordinating or tuning the individual micro-level ant behavior.

Inspiration by the behavior of ants in their colony, the AACA was developed. The AACA uses pheromones trails to allow ants to 'sense' and move towards similar ants within the world, hereby effectively overcoming their relatively limited one-cell sensorial perception. The picking-up and dropping probabilities are both function of *f that* converts the average similarity of data objects into the probability of picking-up or dropping for an ant. To further improve the performance of the algorithm, we propose to substitute Equations 1 and 2 of pick up probability, p_p, and Equation (1) of dropping probability p_d by equations 2 and 3 as follows:

$$p_p(i) = \frac{1}{\tau(i)} \cdot \left(\frac{k_p}{k_p + f(i)} \right)^2 \qquad (3)$$

$$p_{d(i)} = \frac{1}{\tau(i)} \cdot \left(\frac{f(i)}{k_d + f(i)} \right)^2 \qquad (4)$$

Where *f*(*i*) is the density dependent function, *τ(i)* is the quantity of pheromone in the current position *i*, and k_p and k_d are the picking and dropping probability constants, respectively. Note that a new parameter is introduced in relation to standard ant-based clustering which represents the pheromone level at each position on the grid. According to Eq. (3), the probability that an ant picks up an item from the grid is inversely proportional to the amount of pheromone at that position and also to the density of objects around *i*. This formula thus accounts for the pheromone reinforcement signal in regions of the space filled with similar objects. By the same token, Eq. (4) states that regions with high concentration levels of pheromone are attractive for the deposition of more objects of similar type. It is important to observe that a region with a high quantity of pheromone tends to be either a recently constructed cluster or a cluster under construction for which pheromone evaporation and diffusion procedures are implemented. The pheromone trail updates applied to all couplings is done according to the following equation:

In case of dropping item i

$$\tau\ (i) = \rho.\tau\ (i) + \sum\nolimits_{k=1}^{k=s^2} \tau_k \qquad (5)$$

and in case of pickup item i

$$\tau\ (i) = -\rho.\tau\ (i) + \sum\nolimits_{k=1}^{k=s^2} \tau_k \qquad (6)$$

Where ρ represents the persistence of the pheromone trail. With $0 < \rho < 1$, $(1 - \rho)$ represents the evaporation. The parameter ρ is used to avoid unlimited accumulation of the pheromone trails and allows the algorithm to forget previously done bad choices. τ_k is the amount of pheromone in position *k* of surrounding perception area s^2 of the current position of ant i. AACA initializes the density of the aggregation pheromone on the environment (each grid position in our implementation) to a constant so that new individuals are initially sampled according to a uniform distribution over the entire search space.

B. Improved perception of the environment

In conventional ant algorithms, the value of the density function, *f*(*i*), given by equations 1 and 2, depends on the vision field, s^2, of each ant. The definition of a fixed value for s^2 may sometimes cause inappropriate behaviors, because a fixed perceptual area does not allow distinguishing between clusters of different sizes. A small area of vision implies a small perception of the cluster at a global level. Thus, small clusters and large clusters have drawbacks. The agent only perceives a limited area of the environment. In some problems, the use of a *too restrictive* perception field may be limiting factor while *too broad* vision may cause undesirable merging of groups. On one hand, even if a cluster is perfectly homogeneous (with identical elements) and sufficiently large, there still exists a small probability that an agent picks up a datum from the cluster n_d drops it somewhere else. On the other hand, a large vision field may be inefficient in the initial iterations, when the data elements are randomly scattered on the grid since analyzing a broad area may require analyzing a large number of small clusters simultaneously.

In order to overcome this difficulty, a progressive vision scheme was proposed as follows:

When an ant perceives a 'large' cluster, it increments its perception field (s^2) up to a maximal size. Now, s^2 is a specific parameter for each ant that will be dynamically independently updated while running the algorithm. The question that remains is: 'How can an ant agent detect the size of a cluster so as to control the size of its vision field?'.

Because, the value of *f*(*i*) increases as the clustering proceeds, due to the fact that large clusters tends to be formed, then this relationship can be used to obtain a controlling parameter used to detect the size of constructing clusters as follows:

$$If\ \ f(i) > c\ \ and\ S^2\ \le S^2_{max},\quad then\ S^2 \leftarrow S^2+ns$$

when *f*(*i*) achieves a value greater than a pre-specified threshold *c*, the parameter s^2 is incremented by *ns* until it reaches its maximum value.

C. Improved similarity scaling factor (Adoption)

The performance of the algorithm crucially depends on ***α*** parameter which scales dissimilarities within the neighborhood function *f(i)*. If ***α*** is large, then the similarity between objects increases such that it is easier for ants to drop objects but difficult to pickup objects, hence a smaller number of clusters can be formed easily and it contributes to the formation of coarse clusters. If ***α*** is small, then similarity between objects decrease such that it is relative easy for ants to pickup objects but difficult for ants to drop objects. Hence, a large number of clusters can be formed easily, and it contributes the formation of fine-grained clusters. Therefore, the appropriate setting of α value is important and dependent on the statistical pattern of data. In higher dimensionality, it is found that an appropriate ***α*** value cannot be determined without a-priori knowledge of the data's structure

An automatic adaptation of α can be obtained through the tracking of the amount of activity, which is reflected by the frequency of the agents' successful picking and dropping operations. The scheme for *α*-adaptation for each ant used in the proposed algorithm is that the parameter ***α*** of each ant is updated using the rule:

$$\boldsymbol{\alpha} = \boldsymbol{\alpha} \pm \Delta x, \quad ,\text{where } 0 < \Delta x < 1$$

D. Modified similarity density function *f(i)*

The similarity function *f* of Eq. 1 and 2 is replaced by the a modified version $f^*(i.)$

$$f^*(o_i) = \max\left\{0, \frac{1}{s^2}\sum_{y_b \in S} 1 - (\frac{d(o_{i,} o_j)}{u.v.\alpha})\right\} \tag{7}$$

Where μ divides the similarity function by average, which is calculated as follows:

$$\mu = \frac{2}{N(N-1)}\sum_{k=1}^{N}\sum_{L=1}^{K-1}(d(k,L) \tag{8}$$

μ is the average similarity scaling factor, and *N* is the number of data vectors.

In equation 7 the parameter *V* denotes the speed of the ants which is mainly based on the version proposed by [20]. In that version, all ants move with the same speed. We have introduced another parameter with random speed for each ant. This speed is distributed randomly in [1, v_{max}], where v_{max} is the maximum speed, and the term *V* is replaced by $\frac{v-1}{v_{max}}$.

The fast moving ants are not as selective as slow moving ants in their estimation of the average similarity of an object to its neighbors. The diversity of ants allows forming clusters over various scales simultaneously: fast ants form coarse clusters on large scales, i.e. drop items proximately in the right coarse grained region, while slow ants take over at smaller scales by placing objects with more accuracy.

IV. Evaluation Method

The evaluation of the overall design and the prototypical implementation of the new ant based clustering algorithm described in this work are presented next. We present experimental investigation of the performance of ACAA and its comparison to classical data-mining techniques with a brief description of the experimental setup.

A. Selection the comparative methods

- *K-means:* algorithm is a commonly unsupervised clustering algorithm that solve the well known clustering problem. Starting from a random partitioning.
- *Average link agglomerative hierarchical[3]l* algorithm is the second method, the implementation of this method based on the average link linkage metric is used.

The two algorithms require the correct number of clusters as an input parameter. In order to avoid the introduction of an additional source of error we therefore provide the correct number of clusters to k-means, average link agglomerative clustering and, thus giving the same advantage to the two algorithms.

B. Evaluation function measures

To evaluate the quality of the clustering, we adopted four quality measures widely used in the data clustering literature. By using both external and internal measures we have confidence that our evaluation will be justified. The first internal evaluation measure is *intra-cluster variance* [17][18] which measures how similar are the elements belonging to the same cluster, by computing the sum of squared deviations between all data items and their associated cluster centre. The second internal measure is *Dunn index* , which determines the minimal ratio between cluster diameter and inter cluster distance for a given partitioning. Thus, it captures the notion that, in a good clustering solution, data elements within one cluster should be much closer than those within different clusters. As all experiments are run on benchmark data with the correct class labels known, we have the chance to use external evolution measures. For this purpose we have chosen two external measures. The first is the *F-measure[19]*, which combines the *Precision* and *Recall* ideas from the information retrieval literature. The second is the *Rand index[20]* which has been proposed as a more accurate measure. The *Rand index* penalizes both false positive and false negative decisions during clustering. A week point of the Ran index is its high sensitivity to the number of clusters identified, which make it difficult to compare partitioning with a different number of clusters. this is counterbalanced by the use of *F-measure*, which is less affected by deviations in the number of clusters. The *F measure* in addition supports differential weighting of these two types of errors

V. Experimental datasets

To ensure compatibility, and generality, the datasets employed in our analysis are obtained from two different resources. The first category of our experimental datasets is a real life datasets obtained from the machine learning repository. The second experimental datasets are of a synthetic datasets generated by implemented procedure. Which both have a tradition of use in the general clustering literature and have previously been applied for the evaluation of ant-based clustering [9]. Most existing work on clustering analysis uses both synthetic data and real life data to test validity and performance of the proposed algorithms.

A. Syntactic data sets

We created a data generator to produce large synthetic data sets. The extent of a cluster in each of its dimensions can be controlled and so also the max and min of each dimension enabling generation of arbitrary shaped clusters. The generated dataset has been frequently employed in the literature on evaluation and testing new clustering techniques. The generated datasets are two-dimensional and consist of four clusters of equal size, eight, and sixteen clusters of different size. All data sets are generated using Gaussian distributions ($\overrightarrow{\mu}$, $\overrightarrow{\sigma}$) with parameters mean $\overrightarrow{\mu}$ and standard deviation $\overrightarrow{\sigma}$ .. The structure of each cluster is provided by a cluster prototype, simply a definition of the mean $\overrightarrow{\mu}$ of the members held in that respective cluster. Each dataset holds a total of 800 samples that are located in two dimensions. All datasets are generated using $\overrightarrow{\sigma}$ = [2, 2] in both dimensions, with the exception of square 0, where $\overrightarrow{\sigma}$ = [1, 1]. The distance between the centers of the clusters, ΔC, and the number of samples allocated in a cluster, NC, is subject to modulation. This gives rise to two families of synthetic datasets, square[0. . . 7] and sizes[1. . . 5], described in the following section. The modulation of these features allows to examine the robustness of the algorithm on datasets with overlap between the samples and unequally sized clusters.

1.Datasets: *square [0 ... 7]* the square datasets are varied with regards to the means $\overrightarrow{\mu}$ of the four. This gives rise to an increase in overlap between the distinct clusters. With each cluster being constituted by 200 samples, all are equally weighted. The parameters of the individual sets contained in this family are summarized neatly in the top rows of Table 1.

2.Datasets: *size:[1 ...5]* the sizes datasets impose unequal weights to the clusters, while retaining the same inter-cluster distance over all sets. Instead of using a constant of 200 samples for each cluster, the number of samples constituting one of the clusters is gradually increased.

B. Real data set from machine learning

Our experiments were conducted on a number of well-known datasets considered as the best known database to be found in the pattern recognition literature which can be loaded from UCI machine learning database.

TABLE 1: Summary of generated synthetic datasets with different properties, C_i gives the number of clusters and Ni gives the number of data elements for cluster *i*.

dataset name	C1				C2				C3				C4				C_i	N_i
	x1_mean	y1_mean	x1_stdev	y1_stdev	x2_mean	y2_mean	x2_stdev	y2_stdev	x3_mean	y3_mean	x3_stdev	y3_stdev	x4_mean	y4_mean	x4_stdev	y4_stdev		
square 1	0	0	2	2	10	10	2	2	0	10	2	2	10	0	2	2	4	4x200
square 2	0	0	2	2	9	9	2	2	0	9	2	2	9	0	2	2	4	4x200
square 3	0	0	2	2	8	8	2	2	0	8	2	2	8	0	2	2	4	4x200
square 4	0	0	2	2	7	7	2	2	0	7	2	2	7	0	2	2	4	4x200
square 5	0	0	2	2	6	6	2	2	0	6	2	2	6	0	2	2	4	4x200
square 6	0	0	2	2	5	5	2	2	0	5	2	2	5	0	2	2	4	4x200
square 7	0	0	1	1	4	4	5	5	0	4	2	2	4	0	5	5	4	4x200
size 1	0	0	2	2	10	10	2	2	0	10	2	2	10	0	2	2	4	200,100,100,400
size 2	0	0	2	2	10	10	2	2	0	10	2	2	10	0	2	2	4	150,100,300,250
size 3	0	0	2	2	10	10	2	2	0	10	2	2	10	0	2	2	4	150,150,100,100,
size 4	0	0	2	2	10	10	2	2	0	10	2	2	10	0	2	2	4	50,150,100,400
size 5	0	0	2	2	10	10	2	2	0	10	2	2	10	0	2	2	4	50,50,450,250

Four collections were selected from the repository of the real data sets. Although they are small in size, they have irregular cluster distribution, which is an important factor for testing the usability of the proposed algorithm Collections compare our results against other classical clustering algorithms.(see Table 2).

1. *Iris* data set, this is perhaps The Iris data contains 150 items described by 4 attributes. Its 3 clusters are each of size 0. Two of them are linearly non-separable.
2. *yeast* data set is a Probabilistic Classification data for Predicting the Cellular Localization Sites of Proteins contains 1484 data elements described by 8 attributes.
3. *Breast Cancer Wisconsin* data, They describe characteristics of the cell nuclei present from a digitized image of a fine needle aspirate. It contains 699 items

described by 9 attributes. Its 2 clusters are of size 458 and 241 respectively.

4. *Zoo* dataset is A simple database containing 17 Boolean-valued attributes. The "type" attribute appears to be the class attribute. Here is a breakdown of which animals are in which type. This dataset contains 100 animals, each of which has fifteen Boolean attributes.

C. Data preprocessing

All types of these data are normalized in each dimension. For the ant-based algorithm and average link agglomerative clustering, pairwise dissimilarities are precompiled and normalized, K-means require the computation of distances between data items and cluster representative which do not necessarily correspond to data items and possible change in each iteration), such that a pre-computation of distances is not possible for this algorithms. This clearly involves an additional computational overhead for K-mean method, which rises with increasing dimensionality of talked data. We have include two different distance functions: one the Euclidean distance which gives us the magnitude of difference between the two vectors, the second, cosine function which gives us a measure of how similar two vectors are.

TABLE 2. Summary of the used real data sets from Machine Learning Repository.

Dataset name	**Cs**	Number of data elements for **C**$_i$	**N**	**type**
Iris	3	3x15	150	Real
Breast Cancer Wisconsin	2	458 , 241	569	Integer
Yeast	10	463, 429 , 244 ,163, 51, 44, 37,30 ,20, 5	1484	Real
Zoo	7	41 , 20 , 5 , 13 , 4 , 8 , 10	100	Boolean

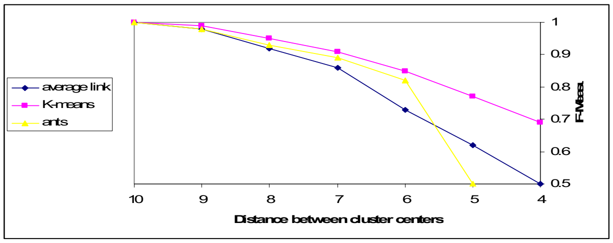
VI. Experimental results

Results obtained are discussed in term of sensitivity to clusters overlapping, sensitivity to clusters sizes, comparison with benchmarks and runtime

A. Sensitivity to overlapping clusters

On the synthetic datasets we have used a relative criterion (F-measure) allowing us to study the sensitivity to over lapping clusters. The performance of the three algorithms has to decreasing with a shrinking distance between the clusters, as points with the region of overlap cannot be correctly classified. Figure 2 illustrates a plot of the algorithms performance (as reflected by the F-measure) versus the distance between the neighboring clusters. A number of trends can be observed in this graph. It is obvious from the graph shown that *K-means* is a very competitive algorithm and has showed best results. *Average link agglomer*ative clustering generally shows poor results and has trouble in identifying the principle clusters while the results of ant-based algorithm are very close to those for *K-means* on the simplest dataset, but its performance drops slightly. Still, it performs scientifically better than average link on the first five test sets. Also the ant algorithm reliably identifies the correct number of clusters on the first four data set.

Figure 2: Performance as a function of the distance between the cluster centers



B. Number of clusters

The AACA was tested on the synthetic that have various cluster structures exist within the data, the results characterized by the ability to identify the number of clusters in the collection that is processed. The performance of the AACA is superior to these with clear cluster structure. The algorithm is quite reliable at identifying the correct number of clusters in the case of the clusters is spatially well or in the case of clear density gradients as shown in the squae1 dataset. On another hand, the majority of popular methods (*K-means, average link*) require the input parameter that constitutes the number of outcome groups. The performance of both *K-means* and agglomerative algorithms perform very badly on this data. The two algorithms are equally failing to correctly identify the clusters in spite of the fact that they have a priori knowledge of the correct number.

However, the ant based clustering has not been provided with the correct number of clusters, in order to get more precise idea of the performance of ant-based clustering, we therefore additionally analyze its success at identifying the correct number of clusters in the data. The experimental results shown that ant-based clustering performs very well, it is much less affected by the lack of spatial separation between clusters than other classical algorithms.

C. Sensitivity to differing cluster sizes

The analysis of the results shows that the AACA is characterized by the proportional distribution of elements among clusters. Also, the trend of creating one superior group can be noticed. In this set of experiments the sensitivity to different-sized clusters is studied using the synthetic datasets size1 to dataset size5. The results of AACA processing are much better than results for K-means processing. It is quite important to observe that the AACA clustering method has a tendency to limit the effect of creating one superior group instead of creating more balance clusters with high degree of cohesion. The average link method gives much worse results than the first two methods. Average link method has a tendency to create one predominant group. The performance of AACA is superior and performs very well on these data sets as reflected by the F-measure (Figure 3). AACA, in fact it is hardly affected at all by increasing deviation between cluster sizes.

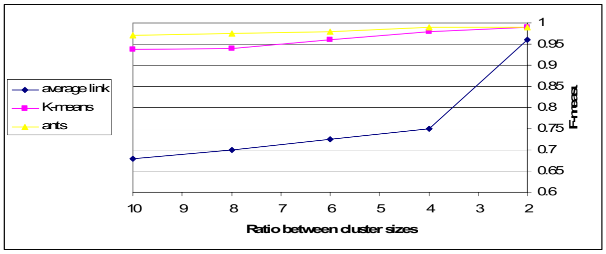


Figure 3: performance as a function of the ratio between cluster sizes on the sizes datasets by using F-measure.

D. Cluster analysis on benchmark real dataset

The first observation on this data tells us, that the results obtained on synthetic data are more accurate than results on the real data. The obtained results show that there is no uniform picture to decide which algorithm is better than another. Since the performance of this data depends on the actual structure with the data (classes are not well-separated), AACA has found trouble to identifying the number of clusters on Zoo, and Yeast data sets and this effects to the obtained values under External measures. However, the bad-separation for this real data set the results poor performance K-means and average link algorithms.

According to the current definition of neighborhood function, Ant-based clustering requires clusters to have a minimum size to construct a stable clusters on the grid which results to the second weakness of the ant-based clustering when it works on small cluster size (Zoo data show The minimum size of clusters). While this is a clear limitation of the algorithm, it could be overcome through the use of modified neighborhood function.

Generally the results obtained by AACA are very good when comparing to the other Algorithms, the degree of cluster compactness and separateness in AACA are obviously improved. The comparative results indicate that AACA generates more compact clusters as well as low error rates than the other clustering algorithms. It is worth mentioning that, different from its competitor, ant clustering algorithm has not been provided with the correct number of clusters.

E. Run Time Performance

The experiments show that for small collections of datasets, the AACA method is slower than other tested methods. However, it should be noticed that with an increase in the dimensionality and the bigger size of collection presented method tends to be ahead of the competitors. average link method and AACA are able to return results faster than the K-means, at the same time the quality and group distribution is much worse of average link method than AACA, where as K-means starts to have convergence problems for the increase of dimensionality and size of collections. This causes K-means to exceed the upper limit of iterations (2000 in our implementation) in almost all runs. Which results in excessive runtimes? Figure 19 presents time of processing for datasets collections with different sizes. The time of processing depends on the number of resultant groups. The results with the best quality and good speed are obtained only by AACA. It is also important to mention that the fastest results are generated using quite small group of ants. It is connected with loss of quality but even so the results are still better than results obtained by other methods.

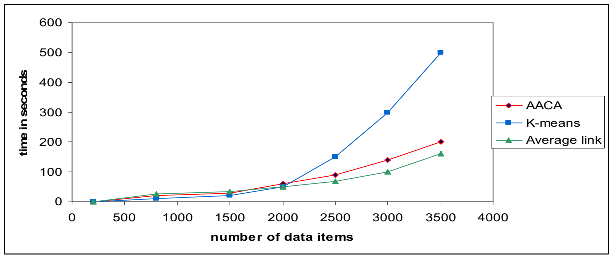


Figure 4. Relation between time and number of proposed dataitems.

VII. Conc;usions

In this paper, we have presented a new ant-based algorithm named AACA for data unsupervised clustering and data exploratory analysis. The proposed algorithm has been tested on many data sets and compared with other clustering algorithm available in the literature. It has also been shown that the proposed algorithm, with the new modifications, out performs other algorithms in term accuracy and run time.

Tests performed in test environment proved the utility and advantages of method created in this paper. The results obtained during experiments are characterized by good quality, speed for big collections of dataset collections and flexibility in determining the number of resultants groups. It seems that there is a possibility to increase the performance of calculations by implementing a parallelization in processing. Most other clustering methods (like K-means an Average link agglomerative clustering) rely on the specification of K as an input parameter. This requires a priori knowledge or interaction with another algorithm, which can be problematic, as current methods for automatic determination of number of clusters in a data set are rather limited.

On the other hand, impossibility to directly define the number of resultant clusters can be recognized as a disadvantage of the ant-based clustering algorithms. There are many application in which user requires the ability to define that value by himself. In ant-based clustering, there is currently no possibility to identify the k clusters that are to be generated.

Seen purely as a clustering algorithm, ant-based clustering performs well in our comparison to the popular methods of K-means, agglomerative hierarchical clustering.

References

[1] Murtagh, F. "A survey of recent advances in hierarchical clustering algorithms," *Computer Journal*, vol. 26, 4,pp: 354-359, 1983.

[2] M. Eisen, P. Spellman, P. Brown, D. Botstein, "Cluster analysis and display of genome-wide expression patterns," PNAS vol. 95, no. 25, pp: 14863–14868, 1998.

[3] Liew, A.W.-C.; Hong Yan; Law, N.F.,"Image Segmentation Based on Adaptive Cluster Prototype Estimation," *IEEE Trans. on Fuzzy Systems,* vol. l., no. 4, pp:444 - 453 Aug. 2005

[4] Watanabe, S., Minami, Y., Nakamura, A., Ueda, N.," Variation Bayesian estimation and clustering for speech recognition,**"** *IEEE Trans. on Speech and Audio Processing,* vol.12, no.4, pp:365 - 381, July 2004

[5] Sadri, J., Suen, C.U., Bui, T.D.,"A New Clustering Method for Improving Plasticity and Stability in Handwritten Character Recognition Systems,**"** *In proc***.** ICPR'06,2006, vol. 2, pp:1130 - 1133.

[6] Ye Qian," Two Stage Fuzzy Clustering Based on Latent Knowledge Discovery and Its Application in the Credit Market,**"** *in proc***.** ICARCV '06, 2006, pp: 1-5.

[7] Vlajic, N., Card, H.C., "Vector quantization of images using modified adaptive resonance algorithm for hierarchical clustering," *IEEE Trans. on Neural Networks,* vol. 12, no. 5, pp:1147-1162, Sept. 2001.

[8]Chonghui Guo, Hongfeng Jia, Na Zhang,"Time Series Clustering Based on ICA for Stock Data Analysis," in proc. WiCOM '08,2008, pp:1-4.

[9] A. K. Jain, M. N. Murty and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264-323, 1999.

[10] Rodrigues, P.P.; Gama, J.; Pedroso, J.P., "Hierarchical Clustering of Time-Series Data Streams,**"** *IEEE Trans. on Knowledge and Data Engineering*, vol.20, no.5, pp:615 - 627, May 2008.

[11] Masulli, F.; Rovetta, S.,"Soft transition from probabilistic to possibilistic fuzzy clustering," *IEEE Trans. on Fuzzy Systems,*vol. 14, no.4, pp: 516 - 527, Aug. 2006.

[12] A. Colorni, M. Dorigo, V. Maniezzo *et al.* "Distributed optimization by ant colonies," in *Proc. 1st Eur. Conf. Artif. Life*, 1991, pp: 134–142.

[13] R. Eberhart, Y. Shi, and J. Kennedy, *Swarm Intelligence*. San Francisco, CA: Morgan Kaufmann, 2001.

[14] Deneubourg, J.L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C. & Chétien, L. (). "The dynamics of collective sorting: robot-like ants and ant-like robots," *Proceedings of the 1st International Conference on Simulation of Adaptive Behaviour*, MIT Press, Cambridge, MA, pp. 356-363, 1991

[15] Chrétien, L., "Organisation Spatiale du Matériel Provenant de l'excavation du nid chez Messor Barbarus et des Cadavres d'ouvrières chez Lasius niger (Hymenopterae: Formicidae)", Ph.D. dissertation, Université Libre de Bruxelles, 1996.

[16] E. Lumer and B. Faieta, "Diversity and adaptation in populations of clustering ants," *in Proc. Third International Conference on Simulation of Adaptive Behavior: From Animals to Animats* 3,J.-A. Meyer and S. W. Wilson, Eds., MIT Press, Cambridge, MA, pp. 501–508, 1994

[17] B. Ghosh-Dastidar and J.L. Schafer, "Outlier Detection and Editing Procedures for Continuous Multivariate Data,". *ORP Working Papers*, September 2003.

[18] Ester, M., Krieggel, H-P., Sander, J. and XU, X. "A density-based algorithm for discovering clusters in large spatial databases with noise," In Proc. of the 2nd ACM SIGKDD, pp: 226-231, 1996

[19] SCOTT, D.W. 1992. Multivariate Density Estimation. Wiley, New York, NY.

[20] Handl, J. and Meyer, B., "Improved ant-based clustering and sorting in a document retrieval interface," *PPSN VII, LNCS* 2439,2002.