ANT-BASED CLUSTERING: A COMPARATIVE STUDY

Gyanesh Das*, Dr. Bijay Kumar Nanda

***Corresponding author: gyanesh_das@rediffmail.com**

Abstract. Ant-inspired clustering and sorting is a nature-inspired heuristic for general clustering tasks. Clustering task aims at the unsupervised classification of patterns in different groups. Finding clusters in data is a challenging problem. Clustering with ant based algorithms is most promising method has been shown to produce good results in a wide variety of problems. In this paper, we present a survey on ant-based clustering techniques. We expect that a study of ant colonies can provide new insights for clustering techniques.

Keywords: clustering, nature inspired, unsupervised classification

1. Introduction

In the last two decades, many advances on the computer sciences have been based on the observation and processes of the natural world. The study of ant colonies has offered great insight in this aspect. An ant colony has many characteristics that are considered useful. It is composed of many agents which, although simple individually, can perform rather complex tasks as a group, but without central coordination. Some examples are building an ant nest, brood pits and cemeteries, hunting and foraging food. The high number of individuals and the decentralized approach to task coordination means that ant colonies show high degrees of parallelism, self-organization and fault tolerance. All of which are desired characteristics in modern computer systems. In this survey, we'll concentrate on ant-systems for the clustering problem. Cluster analysis involves grouping data objects into clusters, having within a cluster (intra-cluster) similarity and between clusters (inter-cluster) dissimilarities optimised [1].

Ant-based clustering and sorting [2] was inspired by the clustering of corpses and larval sorting activities observed in real ant colonies [3].

2. Ant-Based Techniques

In this work, Ant Colony Optimization a group of ant-agents leaves a chemical, pheromone through the path which they have used. Shorter paths will leave stronger pheromone. The next ants, when deciding which path to take, tend to choose paths with stronger pheromone with a higher probability, so as shorter paths are found; more ants try to explore these paths, by a positive reinforcement cycle.

This technique demonstrated good results for the combinatorial problems, and was eventually generalized for any problem that could be reduced to a search on a graph [2]. This led to the popularization of the technique, So that today ant inspired technologies are active areas of research. One important characteristic of the ant-inspired technologies is shown on the work on network routing by Ant Colony Optimization, ANT NET [4]. In this work, after the system finds the optimal routing, it is able to quickly adapt to changes of the environment.

The root of this adaptation is the stigmergic nature of the ant-system. Stigmergy is a central idea of all ant-based algorithms. Here, it happens, as described before, by the leaving of pheromone" trails. The pheromone value of the path that composes the optimal solution is higher than that of the non-optimal solutions after the end of the algorithm. If there is any change in the topology, like a route that fails or a new route, the system can use the existing values of the pheromone trails to adapt to the changes while online. This self-healing, auto-organisable behaviour led to the even further popularization of ant-based techniques. Today we can see the use of this meta-heuristic in applications as diverse as building of hardware numeric operations [6] and clustering [7, 8, 9].

3. Clustering Analysis

Clustering is considered as an unsupervised classification process. Clustering is the process of grouping objects into clusters such that the objects from the same clusters are similar and objects from different clusters are dissimilar. Cluster analysis is also viewed as a tool for exploring the structure of data. The relationship is often expressed as similarity or dissimilarity measurement and is calculated through distance measure. Clustering is used as a data processing technique in many different areas, including artificial intelligence, bioinformatics, computer vision, data mining , data compression, image analysis , image segmentation, information retrieval, machine learning, pattern recognition, statistics and web mining. Cluster analysis is a difficult problem because of many factors (such as effective similarity measures, criterion functions, algorithms, and initial conditions) come into play in devising a well tuned clustering technique for a given clustering problem. Moreover, it is well known that no clustering method can adequately handle all sorts of cluster structures (shape, size, and density). Sometimes the quality of the clusters that are found can be improved by pre-processing the data. Another common technique is to use pre processing steps to fix up the clusters that have been found. The clustering structure can be shown in following figure-1

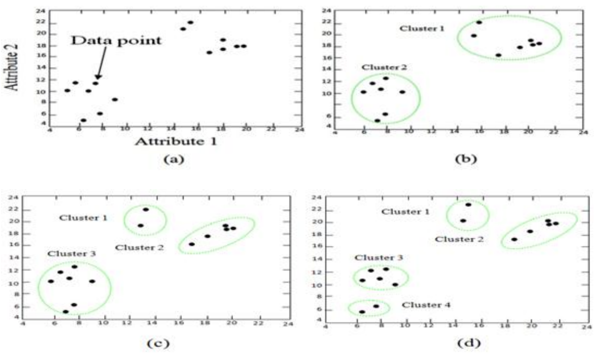

Figure-1. A Clustering Example (a) Dataset, (b) with two Clusters, (c) with three Clusters, (d) clustering results with four nos.

A clustering technique can be broadly divided into three main types: overlapping in other words non exclusive, partitional, and hierarchical. The last two are related to each other in that a hierarchical clustering is a nested sequence of partitional clustering each of which represents a hard partition of the dataset into a different number of mutually disjoint subsets. A hard partition of a data set $X = \{x_1, x_2, \ldots.., x_N\}$ where x_j $(j = 1,2,3, \ldots., N)$ stands for a d-dimensional features, an attribute vector is a collection $C = \{c_1, c_2, \ldots \ldots., c_K\}$ of K non overlapping data subsets $c_j \neq \emptyset$ $(non\ null\ cluster)$ such that $\bigcup_{i=1}^{K} c_i = X\ and\ c_i \cap c_j \neq \emptyset\ for\ i \neq j$. If the condition of mutual disjunction $c_i \cap c_j \neq \emptyset\ for\ i \neq j$ is relaxed then the corresponding data partition can be soft (each object fully belongs to one or more clusters) or fuzzy (each object belongs to one or more clusters to different designs).

How we choose the pattern and classify the given data, as well as how we group the results given a classification is dependent on the use we plan for those results. Thus, there is a wide range of clustering techniques to choose from, and selecting the appropriate combination of methods for a given application is an important part of the clustering process itself.

We can divide the process of cluster analysis into three steps: Feature extraction, Similarity computation and grouping[3]. Feature extraction means selecting which features will be used for organizing the data, and how they should be compared. Similarity computation is the process of taking n members of the data group, and comparing them related to these chosen features. Grouping is determining some sort of structure for the data based on the results of the similarity computation. Most clustering methods involve only the two last steps of the process, but some involves all of the three.

According to their characteristics, clustering techniques can be classified as follows: Hierarchical clustering creates a tree of clusters, where clusters near the root of the tree are more generic, and those near the leaves are more specific. Non-Hierarchical clustering creates clusters that do not possess an explicit relation to each other. Agglomerative clustering creates the clusters bottom-up, starting from individual data and joining them together, while Divisive clustering works top-down, starting with a single cluster and breaking it up. Hard clustering puts each data unit in only one cluster while Soft (or fuzzy) clustering can put the data unit in multiple clusters. Un-supervised clustering performs the clustering without knowledge about the data domain, while supervised clustering uses of existing knowledge about how the data should be clustered, like the number and/or kind of clusters. Beyond the traditional classes there are several clustering techniques that are categorized into independent classes. A short view of these methods is described below.

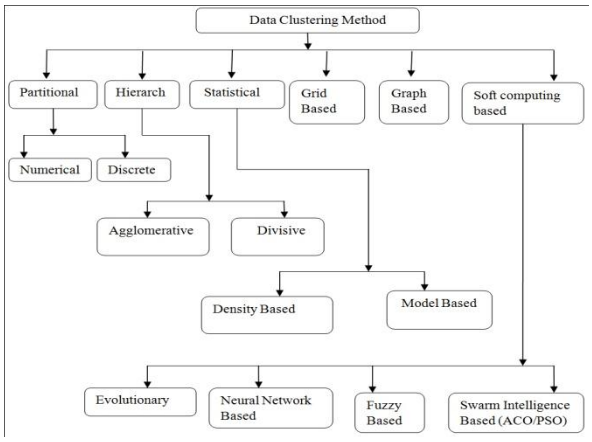

Figure-2. Variours Data Clustering Methods.

K-means is a popular divisive non-hierarchical clustering method, with good performance, but which requires previous knowledge about the number of clusters. C-means is the soft clustering version of K-means. SOM are also divisive non-hierarchical clustering methods like K-means, but they use neural networks to perform the similarity computation step. Ant-based clustering algorithms, as a general rule, can be considered as non-hierarchical, hard, agglomerative clustering methods.

4. Ant Colony for Clustering

Ant-based clustering algorithms are based upon the brood sorting behavior of ants. Larval sorting and corpse cleaning by ant was first modeled by Deneubourg et al. (Deneubourg et al., 1991), for accomplishing certain tasks in robotics. Their work was actually focused on clustering objects by using group of real- world robots, and is known as basic model. This model can be described as follows: The data items are randomly scattered into a two-dimensional grid. Initially, each data object that represents a multi-dimensional pattern is randomly distributed over the 2D space. Each ant moves randomly around this grid picking and dropping the data items. The decision to pick up or drop an item is random but is influenced by the data items in the ant's immediate neighborhoods. The probability of dropping an item is increased if ants are surrounded with similar data in the neighborhood. In contrast, the probability of picking an item is increased if a data item is surrounded by dissimilar data, or when there is no data in its neighborhood. In this way, clustering of the elements on the 2D grid is obtained. Real ants build clusters starting from randomly located corpses. The probability of picking an item is given by:

$$P_p = \left(\frac{k_1}{k_1+f}\right)^2 \qquad (1)$$

where *Pp* is the probability of picking, f is the perceived fraction of items in the neighborhood of the ant, and k_1 is a threshold parameter. The probability of dropping an item is given by:

$$P_d = \left(\frac{f}{k_2+f}\right)^2 \qquad (2)$$

where *Pd* is the probability of dropping, and k_2 is another threshold constant.

Lumer and Faieta (Lumer et al., 1994) have modified Deneubourg et al.'s (Deneubourg et al., 1991) basic model using a dissimilarity-based evaluation of the local density in order to make it suitable for data clustering and it has subsequently been used in data mining (Fraietas, 1995). This algorithm is called LF model or standard ant clustering algorithm (SACA). In this algorithm, each ant-like agent cannot communicate with each other, and they can only sense the similarity of the objects in their immediate region. Lumer and Faieta have introduced the notion of short-term memory within each agent. Each ant remembers a small number of locations where it has successfully dropped an item. And so, when picking a new item this memory is consulted in order to bias the direction in which the ant will move. Thus, the ant tends to move towards the location it last dropped a similar item. Lumer and Faieta define picking up and dropping probabilities as follows:

$$P_p(O_i) = \left(\frac{k_1}{k_1+f(O_i)}\right)^2, \quad (3)$$

$$P_d(O_i) = \begin{cases} 2f(O_i) \; when \; f(O_i) < k_2 \\ 1, \quad when \; f(O_i) \geq k_2 \end{cases} \quad (4)$$

$$f(O_i) = \begin{cases} \frac{1}{i^2}\sum_{o_j \in Neigh(s\times s)}\left[1-\frac{d(o_i,o_j)}{\alpha}\right], when \;\; f > 0, \\ 0 \qquad otherwise \end{cases}$$

Where $f(O_i)$ is a measure of the average similarity of data object O_i with the other data object O_j present in the neighborhood of O_i, $d(O_i, O_j)$ is the dissimilarity between pair of objects (O_i, O_j), α is a factor that defines the scale for dissimilarity, k_1 and k_2 are two constants that play a role similar to k_1 and k_2 in the basic model.
Gutowitz (Gutowitz, 1993) has proposed the complexity seeking ants, which are variants of the basic ants proposed by Deneubourg et al. (Deneubourg et al., 1991). He called the agents as basic ants, which have: 1) A finite memory, which is a register of length 'n' that records the presence or absence of objects at the ant's previous 'n 'locations; 2) An object manipulation capacity; 3) A function that gives the probability to manipulate an object proportionally; 4) To the values in memory and a random variable; and 5) The capability to execute Brownian motion. Although the basic ants have only local perceptual capabilities, they are able to promote global order. The mechanism underlying this phenomenon was carefully investigated by Gutowitz (Gutowitz, 2006). The complexity-seeking ants are allowed to see local complexity and tend to perform action in regions of highest local complexity. Gutowitz (Gutowitz, 1993) has improved on this model by giving the ants, the capacity to sense the complexity (or entropy) of their vicinity. The entropy level of the work area was determined by the presence or absence of objects, so that a place completely empty or completely full would have the lowest entropy, and a check red pattern would have the highest. The level of entropy of the surrounding would affect the propensity of the ants to take an action. In this way, in areas with low entropy the ants would not try to pick or drop anything. These complexity seeking ants were thus able to avoid actions that did not contribute to the clustering process, performing their task more efficiently. Gutowitz (Gutowitz, 1993) has suggested using of spatial entropy to track the dynamics of clustering. The spatial entropy Es at scale 's' is defined by:

$$E_i = \sum_{i\in S} P_i log P_i, \quad (5)$$

where P_i is the fraction of all objects on the lattice that are found in s-patch . Monmarche (Monmarche, 1999) has combined the stochastic and exploratory principles of clustering ants with the deterministic and heuristic of the popular K means algorithm in order to improve the convergence of the ant-based clustering algorithm. The proposed hybrid method is called Ant Class and is based on the work of Lumer and Faieta's Model (Lumer et al., 1994). The Ant Class algorithm allows an ant to drop more than one object in the same cell, forming heaps of objects. Another important contribution of this algorithm is that it also makes use of hierarchical clustering, implemented by allowing ants to carry an entire heap of objects. Slimane, Monmarche, and Venturini (Slimane et al., 1999) have proposed Ant-Class algorithm, which applies explorative and stochastic principles from the ACO meta-heuristic combined with deterministic and heuristic principles from K-means.

Ramos and Merelo (Ramos et al., 2002) have developed a novel strategy called ACLUSTER to tackle unsupervised clustering as well as data retrieval problems. This algorithm was employed for textual document clustering. The authors proposed the use of bio-inspired spatial transition probabilities, avoiding randomly moving agents, which may explore non-interesting regions. In this sense, ants do not move randomly like SACA, but according to transition probabilities that depend on the spatial distribution of pheromone across the environment. If a particular cluster disappears, the pheromone tends to evaporate from that location. This approach is interesting, because pheromone represents the swarm memory and all ants can benefit from it. In other words, the ants share a common memory. Ramos, Muge, and Pina (Ramos et al., 2002) have noticed that the SACA would generate a large quantity of small clusters. They modified the Ant-based clustering by changing the movement paradigm. While the previous works all relied on random moving ants, their ants would move according to a trail of pheromones left on clustering formations. This would reduce the exploration of empty areas, where the pheromone would eventually evaporate. This algorithm was applied to the classification of stone images. They studied the performance of the algorithm on continuous clustering and showed that this improves clustering performance for the ant-based clustering system.

Handl and Meyer (Handl et al., 2003) have applied ant-based clustering as the core of a visual document retrieval system for world wide web searches in which the basic goal is to classify on line documents by contents' similarity. The authors adopted an idea of short-term memory and employed ants with different speeds, also allowing them to jump. In addition, they introduced an adaptive scaling strategy, as well as some further modifications to achieve reliable results and to improve efficiency.

Labroche, Monmarche, and Venturini (Labroche et al., 2002) have proposed a clustering algorithm called ANTCLUST, based on a modeling of the chemical recognition system of ants. This system allows the construction of a colonial odor used for determining the ants' nest membership, such that ants can discriminate between nest mates and intruders.

Kanade and Hall (Kanada et al., 2003) have presented a hybridization of the ant systems with the classical Fuzzy C-Means algorithm (FCM) to determine the number of clusters in a given dataset automatically. In their fuzzy ant algorithm, at first the ant based clustering is refined using the FCM algorithm.

Handl, Knowles, Darigo (Handl et al., 2003), have proposed a scheme that enables an unbiased interpretation of the clustering solutions obtained by ant based clustering algorithms. The authors argue that although many of the results obtained by ant algorithms look promising, there is a lack of knowledge about the actual performance of such algorithms. In order to overcome this limitation, they proposed a technique which is based on the application of agglomerative hierarchical clustering method to the positions of the data items on the grid. Taking into consideration the developed method, the results achieved by the ant-based clustering algorithm proposed by Handl and Meyer (Handl et al., 2002) are compared, using both synthetic and real datasets, with those obtained by two classical algorithms (K-means and agglomerative average link), showing that the ant-based algorithm performs well when compared with them.

Azzag, Monmarche, Slimane, and Venturini (Azzag et al., 2003) have presented a new clustering algorithm (Ant Tree) for unsupervised learning. Tsai et al. (Tsai et al., 2004), have proposed a novel clustering method called ant colony optimization with different favor algorithm (ACODF). In this algorithm, a direct adaptation of the ACO meta-heuristic for solving clustering problems is used. This algorithm performed better than the fast SOM, K-means, and genetic K-means algorithm. Xu et al. (Xu et al., 2004), have presented an artificial ants sleeping model (ASM) and an adaptive artificial ants clustering algorithms(A4C) to resolve the clustering problem in data mining by simulating the behaviors of gregarious ant colonies.

Shelokar, Jayaraman, and Kulkarni (Shelokar et al., 2004) have described an ant colony optimization methodology for optimally clustering N' objects into 'K' clusters. The algorithm employs distributed agents which mimic the way real ants find a shortest path from their nest to food source and back. This algorithm is implemented and tested on several simulated and real datasets. The performance of this algorithm is compared with other popular stochastic or heuristic methods viz. genetic algorithm, simulated annealing and tabu search. The results reveal that the proposed algorithm is effective in terms of quality.

Diego et al. (Diego et al., 2005) have proposed a new version of the Ant-Tree algorithm called Adaptive Ant Tree (AAT), an approach inspired on the self-assembling behavior observed in some species of real ants. The proposed algorithm uses the original Ant-Tree algorithm as the first stage. Then, a re-assignation of ants to different clusters and combination of those clusters is carried out to give a better clustering model. The main features of AAT are given by its capacity of disconnecting ants from the tree under construction without increasing significantly the running time.

Vegard Hartmann (Vegard, 2005) tried a different approach to the ant clustering algorithm by using evolution to train both the system's disparity function and move policy. Each ant would have a neural network which would take the objects of its vicinity as input, and return the move action, and the pick up or drop action, as outputs. By changing the evolutionary system fitness function, he was able to train the ants to create annular clusters; one cluster would be encircling another.

Vizine et al., (Vizine et al., 2005) have proposed an Adaptive Ant-Clustering Algorithm (A2CA). They have made a series of improvements to Lumer and Faieta's (Lumer et al., 1994) system . A2CA is more robust in terms of the number of clusters found and tends to converge into good solution while the clustering process evolves. To achieve these goals, they proposed three main modifications in the SACA: 1) A cooling schedule for the parameter that controls the probability of ants picking up objects from the grid; 2) A progressive vision field that allows ants to 'see' over a wider area; and 3) The use of pheromone function added to the grid as a way to promote reinforcement for the dropping of objects at more dense regions of the grid. These modifications favor an adaptive clustering process, in the sense that the proposed algorithm tends to converge to stable clusters.

Handl, Knowles, and Dorigo (Handl et al., 2006) have described an improved version of the heuristic, called Adaptive Time Dependent Transporter Ants (ATTA), incorporating adaptive heterogeneous ants, a time-dependent transporting activity, and a method that transforms the spatial embedding produced by the algorithm into an explicit partitioning. ATTA is then subjected to the most rigorous experimental evaluation of an ant-based clustering and sorting algorithm undertaken to date. They compared its performance with standard techniques for clustering and topographic mapping using a set of analytical evaluation functions and a range of synthetic and real data collections. Their results demonstrate the ability of ant-based clustering and sorting to automatically identify the number of clusters inherent in a data collection, and to produce high quality solutions.

They have also devised the versions of ATTA such as ATTA-C (ATTA for clustering) and ATTA-TM (ATTA for topographic mapping). This work is followed by the work of Tan et al. (Tan et al., 2006), which removes the ant metaphor from the method and presents a deterministic version of ant-based clustering algorithm.

Yang and Kamel (Yang et al., 2006) have developed a multi-ant colonies approach for clustering data that consists of some parallel and independent ant colonies and a queen ant agent. Each ant colony process takes different types of ants moving speed and different versions of the probability conversion function to generate various clustering results with an ant based clustering algorithm. These results are sent to the queen ant agent and combined by a hyper graph model to calculate a new similarity matrix. The new similarity matrix is returned back to each ant colony process to recluster the data using the new information. Experimental evaluation shows that the average performance of the aggregated multi ant colonies algorithms outperforms that of the single ant based clustering algorithm and the popular K-means algorithm. The result also shows that the lowest outlier's strategy for selecting the current data set has the best performance quality.

Gillne (Gilline, 2007) has investigated the performance of ACLUSTER and ATTA. Under the measures and datasets proposed by Handl et al. (Handl et al., 2006) Based on performance results of both algorithms gathered from numerous runs, the results indicate weaknesses in the design of ACLUSTER while ATTA is well capable of clustering tasks. ATTA is able to detect the correct number of clusters in the data.

Huang, Yang, and Niu (Huang et al., 2007), have described an improved version, called chaotic ant clustering algorithm (CACAS), adopting an important strategy of using chaotic perturbation to improve individual quality and utilized chaos perturbation to avoid the search being trapped in local optimum. The performance of the proposed ant algorithm is compared with the K-means approach and ant based clustering by evaluation functions and topographic mapping using a set of analytical data. The experimental results demonstrate the proposed method is robust and viable.

Zhang et al. (Zhang et al., 2007) have developed an improved clustering algorithm based on ant colony approach. This algorithm is applied in aggregation analysis. The experimental results reveal that the proposed ant algorithm is effective, efficient and quick convergence.

Zahra Sadeghi, Mohammad Teshnehlab and Mir Mohsen Pedram [12] presented a new strategy for clustering using artificial ants in which groups of ants try to do clustering by inserting and removing operations. Clustering is done using groups of ants which are as many as the number of clusters. The goal of each group is to collect members of one cluster. All data objects and ants are spread randomly on the grid. Each ant contains a load list which is initialized with a random object at first. Ants search the grid and try to collect similar data to their load. The load list of ants of each group constructs each cluster. That is, the proposed method uses 'k' ants for finding 'k' clusters and they collect data objects in their load list, so there is no need to the retrieving process. This algorithm outperforms k-means algorithm and LF model of ant clustering. The two new probability functions of picking and dropping are given below:

$$P_{insert}(i) = \begin{cases} \left(\frac{f(i)}{k_{ins}+f(i)}\right)^2 & , if\ f(i) < k_{ins} \\ 1 & , if\ f(i) \geq k_{ins} \end{cases} \quad (6)$$

$$P_{remove}(i) = \begin{cases} 0 & , if\ f(i) < k_{rem} \\ \left(\frac{k_{rem}}{k_{rem}+f(i)}\right)^2 & , if\ f(i) \geq k_{rem} \end{cases} \quad (7)$$

where k_{ins} and k_{rem} are the inserting and removing constants respectively.

Zhao Weili [13] proposed an improved entropy-based ant clustering (IEAC) algorithm. The information entropy to model behaviors of agent is applied. The better quality clusters can be obtained by using the entropy function.

Urszula Boryczka [14] presented a new ant clustering algorithm called ACA for data clustering in a knowledge discovery context. In this algorithm, he employed a modified version of the short-term memory introduced by Lumer and Faieta [15] in order to improve the convergence. The proposed algorithm is evaluated in a number of well-known benchmark data sets.

In this ACA, the following threshold formulae are used for picking and dropping decisions :

$$P^*_{pixl}(i) = \begin{cases} 1\,, if\ f^*(i) > 1 \\ \frac{1}{f^*(i)^2} \quad , else \end{cases} \quad (8)$$

$$P^*_{drop}(i) = \begin{cases} 1\,, if\ f^*(i) \geq 1 \\ \frac{1}{f^*(i)^4} \quad , else \end{cases} \quad (9)$$

where *f*(i)* is a modified version of Lumer and Faieta's neighborhood function:

Lutz Herrmann and Alfred Ultsch [16] presented a new work that shows how the ant-based clustering algorithm from Lumer and Faieta[15] is related to Self-Organizing Maps. They omitted the mechanism of picking and dropping ants for a formal analysis of the underlying formulae. They analyzed the popular technique of Lumer/Faieta. The Lumer/Faieta approach is strongly related to Kohonen's Self-Organizing Batch Map. In their work, a unifying basis is derived in order to assess strengths and weaknesses of both techniques. Finally basic ant based clustering algorithm is given as follows [4]

Algorithm 1. Basic ant-based clustering.

INITIALIZATION
Randomly scatter n items on a 2D toroidal grid
Let G be a population of agents
Each agent in G is randomly assigned (or loaded with) an item
MAIN LOOP
for iteration = 1 to maxiteration **do**
 g := an agent randomly selected from G
 Let the item carried by g be i
 g performs a random move on the grid
 if $p_{drop}(i)$ >Rand[0,1] **then**
  g drops item i at its current location
  PICK := false
  while -PICK **do**
  g moves on the grid randomly until it
  encounters an item q
  if $p_{pick}(q)$> Rand[0,1] **then**
  PICK := true
  g is loaded with q
    end if
   end while
  end if
 end for

It was found that the performance of ant-based clustering is comparable to the k-means technique, when using different cluster quality metrics. Ant-based clustering usually came first or in a close second. When observing the run-time, it was found that ant based clustering for low dimensionality data is slightly slower then k-means but its runtime scales linearly, so that it becomes the fastest algorithm for high-dimensionality d.

5. Conclusions

In this study, we conclude that ant-based clustering techniques are an appropriate alternative to traditional clustering algorithms. Current ant-based clustering has some characteristics that make it desirable; according to the specific problem we are faced. First, it has the ability of automatically discovering the number of clusters. Also, it linearly scales against the dimensionality of data. It automatically generates a representation of the formed clusters that can be intuitively understood by humans. Research on ant-based clustering algorithms is still an on-going field of research. The algorithm has a number of user defined parameters (number of ants, size of the field, movement policies, etc), whose precise effects on the performance are not yet fully known. Work can be done on the analysis of the sensitivity of those parameters. Hartmann proposed [17] to use evolution to determine the move and pick and drop policies, but a similar approach could be used to determine optimal values for other parameters. Ant algorithms in general are known for their self organizing and self healing properties. While so far ant clustering has only been researched to "catch up" with regular clustering, another promising venue of research would be to depart from the traditional techniques and study the possibility of "dynamic clustering", where the data would change during the execution, or new data would be added, and the technique would have to adapt the current solution to the new environment. This is a strong point of ant techniques in general.

References

[1] J.-L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chr´etien. The dynamics of collective sorting: Robot-like ants and ant-like robots. In J.-A. Meyer and S. Wilson, editors, *Proceedings of the First International Conference on Simulation of Adaptive Behaviour: From Animals to Animats 1*, pages 356–365. MIT Press, Cambridge, MA, 1991.

[2] E. Bonabeau, M. Dorigo, and G.Theraulaz. *Swarm Intelligence – From Natural to Artificial Systems.*Oxford University Press, New York, NY, 1999.

[3] A.K. Jain, M.N. Murty, P.J. Flynn, *Data clustering: a review*, ACM Computing Surveys,Vol – 31, No-3, pp. 263–323,1999.

[4] G. Di Caro and M. Dorigo. *"Antnet: Distributed stigmergic control for communication networks".* Journal of Artificial Intelligence, Vol. 9, pp. 317-365, 1998.

[5] Shin Ando and Hitoshi Iba. *"Ant algorithm for construction of evolutionary tree"*. In Proceedings of GECCO 02. IEEE, 2002.

[6] D. Pani and L. Raffo. *"A vsli multiplication-and-add scheme based on swarm intelligence approaches"*. In M. Dorigo et al., editor, Proc. of the 4th Intl. Workshop ANTS, pp. 13-24. Springer Verlarg, 2004.

[7] A. Vizine, L.N. de Castro, E.R. Hruschka, and R.R. Gudwin. *"Towards improving clustering ants: An adaptative clustering algorithm"*. In-formatica Journal, Vol.29, 2005.

[8] Vegard Hartmann. *"Evolving agent swarms for clustering and sorting*". In Genetic Evolutionary Computation Conference, GECCO, Vol. 1, pp. 217-224. ACM press, 2005.

[9] J. Handl, J. Knowles, and M. Dorigo. *"Strategies for the increased robustness of ant-based clustering".* Lecture Notes in Computer Science, Vol. 2977, pp. 90-104, 2004.

[10] E.Bonabeau, M. Dorigo and G. Theraulaz, *"Swarm Intelligence: from natural to artificial systems",* Oxford University Press, Inc., New York, NY, 1999.

[11] M. Dorigo , E. Bonabeau and G. Theraulaz, *"Ant algorithms and stigmergy",* Future Generation Computer Systems, 16(8), 2000, pp. 851-871.

[12] M. Dorigo, G. Di Caro and L.M. Gambarella, *"Ant Algorithms for Discrete Optimization"*, Artificial Life, 5(3), 1999:137-172

[13] Zhao Weili, *"An Improved Entropy-Based Ant Clustering Algorithm",* Proceedings of the 2008 WASE International Conference on Information Engineering, Vol 2, 2009, pp. 41-44.

[14] Urszula Boryczka, *"Ant Clustering Algorithm",* Intelligent Information Systems, Kluwer Academic Publishers, 2008, pp. 377-386.

[15] E.D. Lumer and B. Faieta, *"Diversity and adaptation in populations of clustering ants",* Cambridge: MIT Press, In D. Cliff, P. Husbands, J.-A. Meyer, & S.W. Wilson (Eds.), From animals to animats:

Proceedings of the Third International Conference on Simulation of Adaptive Behavior, 1994, pp. 501-508.

[16] Lutz Herrmann and Alfred Ultsch, *"Explaining Ant-Based Clustering on the basis of Self-Organizing Maps",* ESANN2008 Proceedings, European Symposium on Artificial Neural Networks – Advances in Computational Intelligence and Learning, Bruges (Belgium), April 2008, pp. 23-25.

[17] Vegard Hartmann. *"Evolving agent swarms for clustering and sorting".* In Genetic Evolutionary Computation Conference, GECCO, Vol. 1, pp. 217-224. ACM press, 2005.

[18] Zahra Sadeghi, Mohammad Teshnehlab and Mir Mohsen Pedram, *"K-Ants Clustering – A New Strategy Based on Ant Clustering",* ICS 2008, Tamkang, Vol. 2, 12 Feb. 2009, pp. 329-334.

[19] E. Lumer, B. Faieta, *"Diversity and adaptation in populations of clustering ants",* Proceedings of the Third International Conference on Simulation of Adaptive Behavior: From Animals to Animats 3, vol. 1, MIT Press, Cambridge, 1994, pp. 501–508.

[20] J. Handl, *"Ant-based methods for tasks of clustering and topographic map- ping: extensions, analysis and comparison with alternative methods".* Master's Thesis, University of Erlangen-Nuremberg, Germany, 2003.

Biographical Notes

Prof. Gyanesh Das is currently working in DRIEMS, Cuttack as Assistant Professor in Dept. of Electronics and Telecommunication Engg. He has completed his B.Tech from NIT Bhopal, M.Tech from BPUT, Rourkela and pursuing his Ph.D in ITER, SOA University, Bhubaneswar. His research area is Soft Computing and Channel Equalization.

Prof. (Dr) Bijay Kumar Nanda is presently working in BIET, Bhadrak since last 15 years. He has completed M.Tech in Computer Science from Utkal University, Bhubaneswar and Ph.D. in Computer Science from F.M. University. His research area is Soft Computing and Data Mining.