Journal of Computer Science 9 (6): 780-793, 2013
ISSN: 1549-3636

doi:10.3844/jcssp.2013.780.793 Published Online 9 (6) 2013 (http://www.thescipub.com/jcs.toc)

Hybrid Ant-Based Clustering Algorithm with Cluster Analysis Techniques

[1]Wafa'a Omar, [2]Amr Badr and [1]Abd El-Fattah Hegazy

[1]Department of Information System, College of Computing and Information Technology,
Arab Academy for Science, Technology and Maritime Transport, Heliopolice, Cairo, Egypt
[2]Department of Computer Science, Faculty of Computer and Information, Cairo University,
Arab Academy for Science, Technology and Maritime Transport, P.O Box 2033, Heliopolice, Cairo, Egypt

Received 2013-04-08, Revised 2013-06-04; Accepted 2013-06-07

ABSTRACT

Cluster analysis is a data mining technology designed to derive a good understanding of data to solve clustering problems by extracting useful information from a large volume of mixed data elements. Recently, researchers have aimed to derive clustering algorithms from nature's swarm behaviors. Ant-based clustering is an approach inspired by the natural clustering and sorting behavior of ant colonies. In this research, a hybrid ant-based clustering method is presented with new modifications to the original ant colony clustering model (ACC) to enhance the operations of ants, picking up and dropping off data items. Ants' decisions are supported by operating two cluster analysis methods: Agglomerative Hierarchical Clustering (AHC) and density-based clustering. The proximity function and refinement process approaches are inspired by previous clustering methods, in addition to an adaptive threshold method. The results obtained show that the hybrid ant-based clustering algorithm attains better results than the ant-based clustering Handl model ATTA-C, k-means and AHC over some real and artificial datasets and the method requires less initial information about class numbers and dataset size.

Keywords: Ant-Based Clustering, Clusteranalysis, K-Means, Hierarchical Clustering

1. INTRODUCTION

Swarm intelligence is a scientific field based on observing the natural collective behaviour of social insects (Beekman *et al*., 2008). For example, ant colonies exhibit certain behaviours in nest construction, foraging behaviour, cemetery organization and corpse clustering. Swarm Intelligence (SI) aims to model the simple behaviour of individuals and their local interaction with environment and with neighbouring individuals. The intelligence models seek to find solutions for optimization problems and cluster analysis applications. Cluster analysis is a data mining technology and it employs the similarity measure to differentiate among data objects, so that the objects sharing the highest similarity degree are grouped together.

Cluster analysis is a data exploratory method used in various applications (Mooi and Sarstedt, 2011) such as financial data analysis (Cai *et al*., 2012) and biological data such as clustering gene expression (Nazeer *et al*., 2013). Recently, in the context of data comprehension, researchers have attempted to use SI methods to solve clustering problems. Ant colony clustering algorithms are derived from ant colonies' behavior when constructing cemeteries and sorting corpses. The algorithms have two important features: adopting a distributive process employing positive feedback (Inkaya, 2011) from the ant colony and its environment and creating clusters by projecting high-dimensional attributes into a lower number of dimensions (typically two). However, the algorithm tends to generate more clusters than needed and shows instability

Corresponding Author: Wafa'a Omar, Department of Information System, College of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport, P.O Box 2033, Heliopolice, Cairo, Egypt

regarding the clustering solution (Zhe *et al*., 2011).

The purpose of this research is to improve the performance of the basic ant-based clustering algorithm based on the model of Lumer and Faieta (1994), which operates on cluster analysis applications. Some modifications are introduced to the ant-based clustering algorithm in order to enhance its performance and the resulting clusters. These modifications enable worker ants to consult two matrices by generating a minimum distance link matrix for remembered data objects, or a proximity distance link matrix amongst clusters. In addition, a method is implemented to detect small clusters and to embed their data objects in the closest and the highest resemblance groups. Furthermore, a refinement method is implemented to reappraise intersecting points amongst neighboring clusters to maintain cluster affinity and consistency. Besides these modifications, the research aims to evaluate distance measure as a basic criterion for identifying degrees of similarity amongst data items. Four real datasets and one artificial dataset are used to test the proposed hybrid ant-based clustering algorithm functionality. The resulting clusters are evaluated by four analytical measures: f-measure, Rand index, variance and Dunn index. The outcomes are also compared with the outcomes of adaptive time dependent transporter ants for clustering ATTA-C (Handl *et al*., 2006), k-means and Agglomerative Hierarchical Clustering (AHC) (Mooi and Sarstedt, 2011).

The second part of this study is organized as follows: it discusses cluster analysis algorithms and gives a brief comparison of clustering methods. The third part discusses SI as a collective model and reviews previously used methods of ant-based clustering. The fourth part covers problem statements, modifications to ant-based clustering, algorithm phases, the ants' movement system and a threshold-updating mechanism. In the fifth part, an explanation of the experimental methodology is introduced, including real data sets, parameter settings, analytical measures and experimental results and observations. Finally, in the sixth part, the conclusions of the paper are explained and future tasks are proposed.

2. CLUSTER ANALYSIS

Cluster analysis is a form of data mining which is imposed over a set of objects, with the aim of categorizing data based on a criteria of similarity extracted from information found in the datasets, i.e., the attributes that describe the datasets (Dhiraj, 2009). Many fields benefit from translating underlying datasets into meaningful information, ranging from machine learning, pattern recognition, web mining, textual document collection, image segmentation and areas of economics such as marketing and business (Jain and Maheswari, 2012). The two traditional clustering structures are partition methods that assign a set of objects into non-overlapping clusters such as k-means methods and density-based clustering, while hierarchical methods create a set of sub clusters that are organized into a tree structure, such as AHC (Tan *et al*., 2006).

K-means is a practical and commonly used clustering algorithm for solving clustering problems. The algorithm requires defining the number of partitions, as an input parameter. It groups the objects of a given data set into an optimal partition criterion that minimises a Mean Square Error (MSE) and each delivered cluster is recognised using its centre. K-means generates good results for compact and convex clusters and it functions effectively for clustering large data sets. The computational complexity for traditional k-means is $0(NK\ I_{tr})$, where N is the number of objects, K is the number of clusters and I_{tr} is the number of iterations (Elavarasi *et al*., 2011).

The AHC algorithm starts simply by considering each data object as an individual cluster and then by merging the closest objects into one cluster and repeating the process until no additional objects can be merged. AHC is a common clustering approach since it is applicable to any attribute type. However, AHC does not scale well with large datasets, is sensitive to noise and outliers and its runtime complexity is at least $0(m^2 \log m)$ for m data objects. This makes AHC costly to use for clustering large size datasets (Hastie *et al*., 2009).

In addition, density is a useful feature and it appears in the distribution of the dataset. Density-based clustering generates clusters by locating regions of high density and density-connected objects and separating the identified regions from others of low density, or those defined as noise and excluded from clustering; for example, Density-Based Spatial Clustering of Applications with Noise (DBSCAN). The key features of the algorithm are automatically detecting the number of clusters, discovering clusters of random shapes and successfully defining noise. This method, however, fails to generate a complete clustering (Parimala *et al*., 2011).

The clustering methods discussed so far have achieved positive results in solving clustering problems. However, each method has drawbacks. K-means is inefficient for identifying cluster numbers and optimal

initial partitions, sensitive to outliers and noise and is only applicable to numerical applications (Elavarasi *et al*., 2011). AHC is impractical for working with all sizes of database since its runtime scales up with the growth of the data set sizes, $0(m^2)$ (Hastie *et al*., 2009). DBSCAN is also sensitive to cluster datasets of widely varying densities. Density-based clustering method depends on threshold (τ), which is decided by users. Furthermore, fixed thresholds are not practical to identify similarities among data objects (Parimala *et al*., 2011).

3. SWARM INTELLIGENCE: ANT-BASED CLUSTERING BACKGROUND

Many species live in swarms, an example of which is seen in ant colonies, where ants build complex nests composed of many chambers connected by a network of tunnels. The overarching motivation of ant life is to build and maintain their colonies, which involves many complex activities such as foraging, cemetery organisation and nest construction (Engelbrecht, 2007). Different tasks performed inside or outside the ant colony are accomplished by a sort of local cooperative communication amongst specialised groups of ants, with no leader to direct each individual or assign tasks. SI is a scientific concept used to understand collective behaviour observed in species living in swarms. It is also known as collective intelligence, which means that the collective interactive behaviours amongst an agent group are aimed at problem solving (Martens *et al*., 2011).

A clustering task can be observed in nests of several ant species, where classes of ants organise corpses to form cemeteries or group larvae. A cemetery formation process is similar to a clustering process, where a specialised group of ants collects corpses to form heaps and ants can only share some local information with other ants active in the same area. To do so, each individual ant randomly picks up a corpse, a dead item and drops it in a particular position inside a cemetery based on the physical properties of the corpse. The algorithmic models, which represent swarm collective behavior, are known as Computational Swarm Intelligence (CSI). Ant Algorithms (AA) are emergent, well-known and widely used CSI methods with various models, such as Ant Colony Optimisation (ACO and ant-based clustering (Martens *et al*., 2011).

The first algorithmic model mimicking the foraging behavior of ants is the ant colony optimisation ACO, originally developed by Dorigo and Stutzle (2004). Basic ACO models the ability of ants to find the shortest route between their nest and a food source. Throughout the roaming process, ants deposit pheromone trails as a means of communication, guiding other ants to the preferred paths to follow. Subsequently, many ACO models examples have been developed to mimic the foraging behavior of ants, in order to solve optimization problems such as data mining clustering, classification and feature selection (Jafar and Sivakumar, 2010).

The ant-based clustering algorithm, a type of ACO algorithm, was first introduced for tasks in robotics by (Deneubourg *et al*., 1991). The proposed model was inspired by the clustering of corpses to form cemeteries and the sorting of larvae-important ant colony activities. The basic environment of the algorithm consists of randomly placed high-dimensional data objects, having several attributes in a bi-dimensional grid. The grid size should be large enough to allow ants to roam and search for data objects. The clusters constructed are affected by the original spatial distribution of the objects. The process begins with agents-ants-picking up data objects with low density and similarity. The ants then try dropping the data objects at a suitable location in which similar objects exist already (Blum and Li, 2008).

Later, Lumer and Faieta (1994) extended Deneubourg's basic model of the ant-based clustering algorithm to cluster data of a numerical type. This study extended the algorithm applicability to a wider range of data types and clustering data mining. In Deneubourg's model, data items are labelled as A or B and a cluster number is predefined. The algorithm designed by Lumer and Faieta (1994) attained good rankings compared to other competing algorithms, but it creates small clusters failing to merge with larger clusters (Martens *et al*., 2011). Several solution modifications were introduced, especially for addressing data mining problems such as noise elimination (Zaharie and Zamfirache, 2005) and clustering and topographic mapping (Handl *et al*., 2006). There are also many algorithms to handleclustering problem such as ant-based clustering (Boryczka, 2008), improved entropy-based ant clustering (Weili, 2009), ant-based clustering algorithm proposed (Villwock and Steiner, 2011) and ant-means (Hameurlaine *et al*., 2012).

4. PROPOSED MODEL: HYBRID ANT-BASED CLUSTERING ALGORITHM

4.1. Problem Statements

In this research, a hybrid ant-based clustering model is developed based on the notion of the Lumer and Faieta (1994) model. The proposed algorithm includes four

processes: analysis, clustering, merging and refinement for the purposes of investigating new solutions to the aforementioned cluster analysis challenges by introducing new features to the ant-based clustering algorithm. These features are adapted from classical cluster analysis algorithms: Agglomerative Hierarchical Clustering (AHC) and Density-Based Clustering (DBSCAN).

4.2. Solution Construction

The proposed hybrid ant-based cluster model is composed of four sequential phases: analysis process, clustering process, process of merging and refinement process. The first phase is a learning process that examines a learning dataset. This initial process delivers prime classes or clusters in addition to a prime threshold value that is estimated from a learning dataset. The main task of the second phase is to cluster a test dataset where worker ants pick up data items from the grid environment and attempt to accommodate the data into the most similar classes provided from the analysis process. The proposed algorithm could assign most testing data items to available groups but it creates many small clusters. The small clusters are overcome during the process of merging in the third phase, where small and similar classes are joined together to form new homogenous clusters. In the fourth phase, or refinement process, clusters are improved by eliminating small clusters and relocating their data items into clusters with the most resemblance. Following this, the border of the most neighboring clusters, where border data items can be misclassified, is checked. **Table 1** for general description of the ant-based cluster algorithm.

The functions of ant-based clustering are probability functions (Engelbrecht, 2007) developed by Lumer and Faieta (1994). The probability of picking up and dropping data item Y_a is givenas Equation 1 and 2:

$$P_p(ya) = (\gamma / \gamma + f(ya))^z \quad (1)$$

$$P_d(ya) = \begin{cases} 2f(ya) \text{ if } f(ya) < \gamma \\ 1 \text{ if } f(ya) \geq \gamma \end{cases} \quad (2)$$

The picking up of a threshold for a data vector y_a depends on a similarity coefficient γ or scale factor, a constant value initialised to 0.51 and the local density function $f(y_a)$, Equation 3, defined as follows, where N is the size of the neighbourhood, n is the number of items in squared $N_{n\times n}$ and $d(y_a, y_b)$ is the dissimilarity function Equation 3:

$$f(ya) = \max\left\{0, \frac{1}{n^2}\sum_{ya} \in N_{n\times m}\left(1 - \frac{d(y_a y_b)}{\gamma}\right)\right\} \quad (3)$$

The similarity coefficient γ is dynamically changed at each iteration and it is based on the successful activities of picking and clustering data items by each ant a, $\gamma \in (0,1)$, where $n_f^a(t)$ is the number of failed dropping practices for ant a at time step t Equation 4:

$$\gamma^a(t+1) = \begin{cases} (\gamma^a(t) + 0.01) \text{ if } \frac{n_f^a(t)}{n_f} > 0.99 \\ (\gamma^a(t) - 0.01) \text{ if } \frac{n_f^a(t)}{n_f} > 0.99 \end{cases} \quad (4)$$

In addition, the hybrid ant-based cluster model incorporates some modified features. One method accommodated in AHC is a group average function, which defines the pairwise proximity and average distance among different clusters' pair items (Tan *et al*., 2006). In this model, a minimum group average proximity is favored for the selection of the two closest neighboring clusters. The function is employed to advise ants about which neighboring clusters to examine. Ants need a road map to familiarise themselves with their environment.

The average proximity function operates in the learning, cluster and merging processes Equation 5:

$$\text{proximity}(C_i, C_j) = \frac{\sum_{\substack{x\bullet C_i \\ y\bullet C_j}} \text{proximity}(x,y)}{m_i * m_j} \quad (5)$$

where, C_i and C_i are clusters, which are of size m_i and m_j, respectively.

Another method is inspired by the density-based clustering algorithm. This method posts similar groups of items inside, as well as separated clusters located in the grid environment. During the refinement phase, the main task of ants is to check the border of adjacent clusters to reassess the data items that have fallen within a conjunction area. The steps' processes are briefly described as Rule 1-3:

R1: Compute a mean point $(\bar{x})$ for every cluster.

R2: Find border data items p between two contiguous clusters C_i and C_j:

Table 1. Hybrid ant-based clustering algorithm

Initialisation:	
Scatter data vector, y_a, randomly on a grid	
Initialise parameters γ, α, τ_i& r.	
Phase 1: Analysing learning data vectors $y_{a-N_{learn}}$	
Load Ant-leader-memory ($A_{l\text{-}m,\ m\ =\ 1,\ldots\ldots,mem\text{-}max,}\ y_a$)	
for Ant-leader-memory, $A_{l\text{-}m}$, (m = 1, …….,A_{lm}-max) do	
Compute Closest Matrix ()	Equation 5
Compute Neighbour Function $f(y_a, y_b)$	Equation 3
if ($P_d(y_a) >= \tau_i$) **then**	Equation 2
Drop (y_a, y_b)	
Update Threshold (τ_i, τ_{new}) **end-for**	Equation 6&7
Phase 2: Clustering testing data vectors $y_{a-N_{test}}$	
While (I_{ter} < Max_Iteration)	
Load ($A_{w\text{-}m,\ m\ =\ 1,\ldots,\ mem\text{-}max}$, y_a)	
for all ant-Worker A_w, (w = 1,……., A_w-max) **do**	
Find Minimum Average Distance (y_a, C_i)	
Compute Neighbour Function $f(y_a, C_i)$	Equation 3
if $P_d(y_a) >= \tau$ **then**	Equation 2
Drop(y_a, C_i);	
Update Threshold (τ_i, τ_{new}) **end-for**	Equation 6&7
end-while	
Phase 3: Merging the most similar and neighboring Clusters	
for all C_i, (i = 1,……., C-max) **do**	
Compute Clusters Minimum Average Distance MAD (C_i, C_j)	Equation5
Compute Cluster Neighbour Function $f(C_i, C_j)$	Equation 3
if $f(y_a, C_j) \geq \tau \parallel f(y_b, C_i) \geq \tau$ **then**	
Merge Clusters (C_i, C_j); **end-for**	
Phase 4: Cluster Refinement: Detecting Small Clusters and Checking Boundaries Rule 1-3	
Find Clusters mean points (C_m)	R. 1
for all C_{mi}& C_{mj} , (i &j = 1,……,C_m-max) **do**	
Compute Neighbour Function for Close Clusters $f(C_{mi}, C_{mj})$	Equation 3
if $P_p(C_{mi}) >= \tau$ **then**	Equation 1
Merge Clusters (C_i, C_j); **end-for**	
Find Intersection Vectors for Close Clusters (C_i, C_j)	R. 2
for all y_a, (a = 1,……., y-max) **do**	
Compute Neighbour Function $f(y_a, C_i)$ &$f(y_a, C_j)$	Equation 3
if $P_p(y_a, C_i) > P_p(y_a, C_j)$ && $P_d(y_a, C_j) >= \tau$ **then**	R. 3
Add(y_a , C_j); **end-for**	

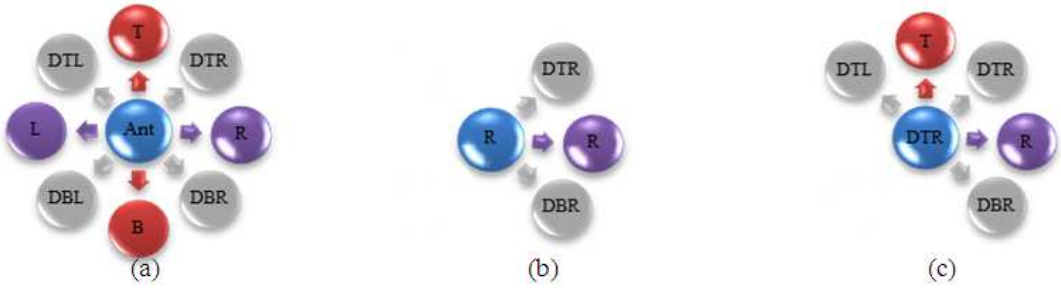


Fig. 1. (a) In the first tour, (b) An ant visits only three new cells and (c) An ant visits only five new cells

for ∀ point $p \in C_{i\text{-}m}$, where m is cluster size.

$$\text{if}(p_x \geq \bar{x}_1 \;\&\; \&p_x \leq \bar{x}_2) \;\&\; \&(p_y \geq \&\; \&\; \bar{y}_1 p_y \leq \bar{y}_2),$$
$$\text{then } p \in \text{Intersection\_points}$$

Note that $(\bar{x}_1 < \bar{x}_2) \& (\bar{y}_1 < \bar{y}_2)$, where (x,y) are vectors of point p, which belongs to cluster C_i and $(\bar{x}_1 < \bar{x}_1) \& (\bar{y}_2 < \bar{y}_2)$ are two-mean vectors for clusters C_1 and C_2, respectively.

R3: Drop a data item p inside a cluster of the highest pickup probability if and only if its response-dropping threshold exceeds the checking threshold.

4.3. New Features Introduced to Enhance Ant-Based Clustering Model

Search and movement system for labour ants is designed. Labour ants, consisting of leader ants and worker ants, search the grid cells to pick up data objects and then drop them near the most similar groups.

Figure 1 shows the search model which designed to allow ants, during their first move to roam and examine their local patch with a radius equal to one and a set of eight possible paths: top, bottom, right, left, top-right, top-left, bottom-right and bottom-left (**Fig. 1a**). For any new step or move, an ant has two possible scenarios. If the ant moves to one of the main directions: top, bottom, right, left, it will visit only three new cells (**Fig. 1b**). Or, if the ant moves to one of the secondary directions-top-right, top-left, bottom-right, or bottom-left-□it will visit only five new cells (**Fig. 1c**).

An adaptive threshold mechanism is designed to find a dynamic threshold value. A threshold is computed from the learned data set during the learning phase, which is utilised by laden ants either to complete the task of pick up, data drop items, or task failure. The initialphase, where leader ants start searching the grid and forming very small clusters, is mainly composed of two data items at a time. The purpose of the slow cluster formation is to allow ants to evaluate and explore the data objects. Therefore, at the analysis phase the determined threshold value τ is initially fixed at 0.8. Once the leader ants complete their first tours by picking up thirty data items, the threshold check τ values is computed by using the average local density function for only the clustered data items. The algorithm repeatedly updates the threshold values at the end of every tour performed by a leader ant by using Equation 6 and 7:

$$\tau_{new} = \frac{\Sigma f(ya)}{\text{no. of clustereddataitems}} \tag{6}$$

$$\tau_{new} = \begin{cases} (\tau + 0.01), & \tau_{new} \geq \tau \\ (\tau - 0.01), & \tau_{new} \leq \tau \end{cases} \tag{7}$$

5. EXPERIMENTAL RESULTS AND OBSERVATIONS

5.1. Experimental Settings

The hybrid ant-based clustering algorithms are executed by using Microsoft Visual Studio 2005 and TANAGRA 1.4.40 on an Intel® CoreTM i5 CPU M 450 @ 2.40GHz, 4GB RAM computer. The algorithm records the results of analytical measures such as cluster numbers, maximum f-measure, Rand index, variance and the Dunn index, besides clustering error and runtime. At the end of the run, the algorithm calculates the mean (μ)and standard deviation (σ) of all the previous evaluation measures. A brief discussion is conducted concerning the datasets, parameter settings, evaluation measures, results and finally a results comparison amongst alternative clustering algorithms.

The hybrid ant-based clustering algorithm is verified by four real numerical datasets from the Machine Learning repository (Frank and Asuncion, 2010) and one artificial dataset, Ruspini (1970). The selected datasets are Iris, Wine, Wisconsin Diagnostic Breast Cancer (WDBC), Vertebral Column_2c and Ruspini. **Table 2** shows the number of records, attributes and distribution of the datasets.

Some parameters need to be set for the hybrid ant-based clustering algorithm. Several parameters of the algorithm are set independently of the datasets. The algorithm uses labour ants, composed of three leader ants and seven worker ants. The ratio of learning and testing datasets is set to 0.5.

The leader ant and worker ant memories are fixed to $LA_{mem} =$ and $WA_{mem = 1}$, respectively; the initial similarity coefficient is set to $\gamma = 0$ the initial threshold is set to $\tau =$ 0.8; and the patch neighborhood is N = 9 cells. r is a radius of perception set to 1. However, other parameters are set based on the dataset size. They are the number of square grid cells, $\sqrt{10 * \text{Nmax}} * \sqrt{10 * \text{Nmax}}$, where Nmax is the number of objects in a dataset; and the total number of iterations is given by Itr = 2000*Nmax.

Table 3 shows two sets of evaluative measures assessing the performance and results of the ant-based clustering algorithm.

Table 2. List of names and features of benchmark real datasets

Dataset	Records	Attributes	Class Distribution
Iris	150	4-Real	3: (50,50,50)
Wine	178	11-Real, 2-Integer	3: (59,71,48)
WDBC	569	30-Real	2: (357,212)
Vertebral Column_2C	310	6-Real	2: (210,100)
Ruspini	75	2-Integer	4: (20,23,17,15)

Table 3. Evaluations measures

Measure function	Equations
F-measure	$\sum_{j}\frac{n_j}{n}\max_i\left(\frac{2\cdot p(i,j)\cdot r(i,j)}{p(i,j)+r(i,j)}\right)$
Rand index	$\frac{FN+TP}{FN+FP+TN+TP}$
Intracluster variance	$\sum_{i}^{C}\sum_{y}^{i}\delta(y,\mu_i)^2$
Dunn index	$\min_{i,j\in C}\left\{\frac{\min_{\mu_i\in C_i,\mu_j\in C_j}[\delta(\mu_i,\mu_j)]}{\min_{l\in C}[diam(C_l)]}\right\}$
Clustering error	$\frac{2}{N(N-1)}\times\sum_{(i,j)\in\{1,\ldots,N\}^2,i<j}\in_{ij}$ $\in_{ij}=\begin{cases}\text{if}(c(o_i)=c(o_j)\ \dot{c}(o_i)=\dot{c}(o_j))\\(c(o_i)\neq c(o_j)\ \dot{c}(o_i)\neq\dot{c}(o_j))\\1\ \text{else}\end{cases}$

The first set consists of the external indices such as number of clusters maximum f-measure and Rand index. The second set consists of internal indices such as variance and the Dunn index. In addition, the number of clusters and cluster error rate are also obtained to measure cluster results.

5.2. Results

The test experiment of the hybrid ant-based cluster algorithm is conducted by running the algorithm fifty times for each of the five real datasets: Iris, Wine, WDBC, Vertebral Column_2C and Ruspini. Results for the hybrid ant-based clustering algorithm are shown in **Table 4**, which displays three main results: the number of clusters, clustering error and duration of all data collections. The outputs are presented in the form of mean and average standard deviation.

The algorithm records the best results finding the exact cluster numbers, with Iris achieving a mean of 2.9800 out of three clusters and Ruspini 4.2600 out of four clusters; while the worst results, recorded in clustering Column_2C, with a mean of 1.0600, only detected one cluster. These findings indicate that the algorithm performs better with linear datasets or semi-correlated classes such as Ruspini and Iris, while its performance degenerates when clustering highly correlated classes such as vertebral column_2C. Clustering error is the second measure in **Table 4**: The least error is 0.0110 obtained for the Ruspini data. The algorithm was able to record 72% zero error for fifty independent runs. The reason for this is that the Ruspini dataset contains separate classes of only 75 objects. In same context, it was observed that the error rate decreases with linear classes such as Ruspini and Iris, since they recorded a low error rate, nearer to zero, in an interval of ($0.01 < \text{errorrate} < 0.13$), while the error rate approached one when clustering correlated classes such as Column_2C (0.8790).

The speed of the algorithm is proportional to the dataset size. The runtime recorded for small Ruspini data was 0.5200, while the runtime for the largest dataset was 2.3200, recorded for WDBC. However, the algorithm was stable in processing a single record for different sizes of the given datasets with an average difference of ∓ 0.0052 ((∑μ_time) / Nmax) / (DatasetNumber). The maximum average distribution equals 0.6462, recorded for the WDBC dataset and the minimum mean distribution, recorded for the Wine dataset, is 0.3736.

Table 5 shows the average results for f-measure and the Rand index, when applying the proposed ant-based clustering algorithm to the Wine and Vertebral Column_2C datasets. The Wine data records f-measure $\mu = 0.8384$ and this indicates an extent of unity of objects inside their clusters. However, the Rand index value $\mu = 0.8402$ records a bit difference due to correlated distribution of data in some area of grid cells or lack of sufficient information at early stage of the clustering process. In general, the algorithm points to successful results for some runs, exceeding the (0.9) value for both f-measure and the Rand index. Vertebral Column_2C is a non-linear dataset and records f-measure ($\mu = 0.7015$) and the Rand index ($\mu = 0.5619$).

In addition, the results for variance and the Dunn index for both WDBC and Ruspini datasets are listed in **Table 6**. The variance for WDBC is 0.8505 which is quite high in terms of the need to minimise the variance value and its distribution is ±0.0698 (**Table 6**). The Dunn index for WDBC is 11.9346 with separation rate comes in range ±0.8616 which reflects a medium differentiation among resulting clusters. The distribution of WDBC is illustrated in **Fig. 2a** and its data objects are gathered into two separate groups composed of two and three clusters. The Ruspini data attains a variance equal to 0.3782 and distribution of 0.0091 (**Table 6**). This implies that most data objects are closely gathered around the mean. In addition, the average Dunn index value for the Ruspini data is 9.6725 with a high separation rate of 4.8174. The algorithm attains the optimum run for Ruspini as it achieves a zero error rate for about 75% of fifty independent runs and the associated variance and Dunn index have values of 0.3740 and 12.7141 respectively. Accordingly, the algorithm successfully assembles similar objects of Ruspini data and groups them into individual clusters (**Fig. 2b**).

5.3. Observations

Experimental results comparing the performance between the proposed hybrid ant-based clustering algorithms and other cluster algorithms including the Handl model, k-means and agglomerative hierarchal clustering algorithms were obtained for five datasets: Iris, Wine, WDBC, Vertebral Column_2c and Ruspini. **Table 7** displays the comparison of the best results among clustering algorithms for fifty independent runs and randomly selected data objects in these datasets.

The results of the Iris dataset are displayed in **Fig. 3a**; it was found that the hybrid ant-based clustering attained the best value of correct clusters with an f-measure of 0.9667 in about 50% of the total number of runs. The Handl model, AHC and k-means algorithms achieved an f-measure of 0.9666, 0.8535 and 0.9600, respectively. The inner cluster compactness, represented as the variance value, was 0.3545 for hybrid ant-based clustering, with a minimum clustering error of 0.0850. The variance values for validity of the Handl model, AHC and k-means are 0.3473, 0.3946 and 0.3467, respectively. The proposed ant-based clustering scores the same clustering error (0.0850) as Handl model, followed by k-means (0.1009); the worst error value is that of AHC (0.2969).

In the same context, **Fig. 3b** shows the Iris results with the best intracluster separation validity; the Dunn index is 9.4986 for the proposed ant-based clustering. The Dunn index values for the Handl model, AHC and k-means are 9.4047, 8.8304 and 9.3456, respectively. **Fig. 3b** shows the best similarity degree, which is the Rand index, for the Iris data set, which is 0.9578 for both the ant-based clustering and Handl models. The Rand index values for AHC and k-means are 0.8625 and 0.9499, respectively.

Deduced from **Fig. 4a** for the Wine dataset, the proposed ant-based clustering achieved the highest intra-cluster variance value (0.7714) but recorded the lowest cluster error value (0.2252) compared to the AHC and k-means algorithms, whose variance values were 0.7557 and 0.7492 and clustering error values were 0.2460 and 0.2369, respectively (**Fig. 4b**). The Handl model attained the lowest intracluster variance (0.2414). Nevertheless, it failed to provide correct groups of clusters where the cluster error was too high (0.899) as illustrated by the red dots in **Fig. 4b**.

Table 4. Results of proposed ant-based clustering

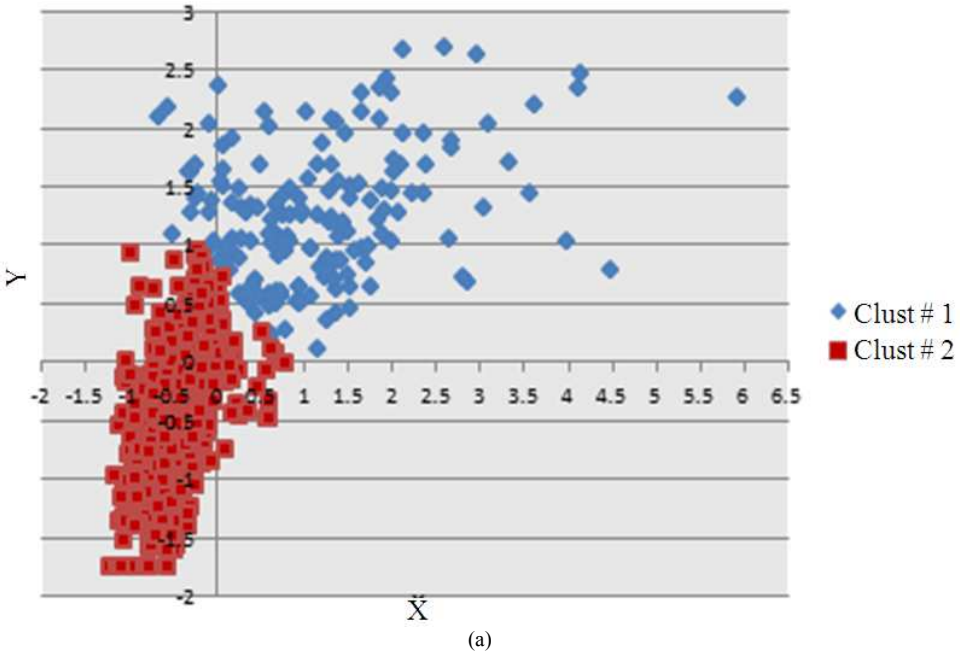
	Clusters found		Clustering Error		Time	
Datasets	μ	σ	μ	σ	μ	σ
Iris	2.9800	0.2441	0.1326	0.0732	0.7400	0.4386
Wine	3.9800	0.7613	0.3213	0.0579	0.9800	0.3736
WDBC	2.3600	0.4800	0.3629	0.0504	2.3200	0.6462
Vertebral Column_2C	1.0600	0.2375	0.8790	0.0093	1.4000	0.5657
Ruspini	4.2600	0.4386	0.0110	0.0181	0.5200	0.5381

Table 5. Comparison of wine and vertebral column_2C by external measures

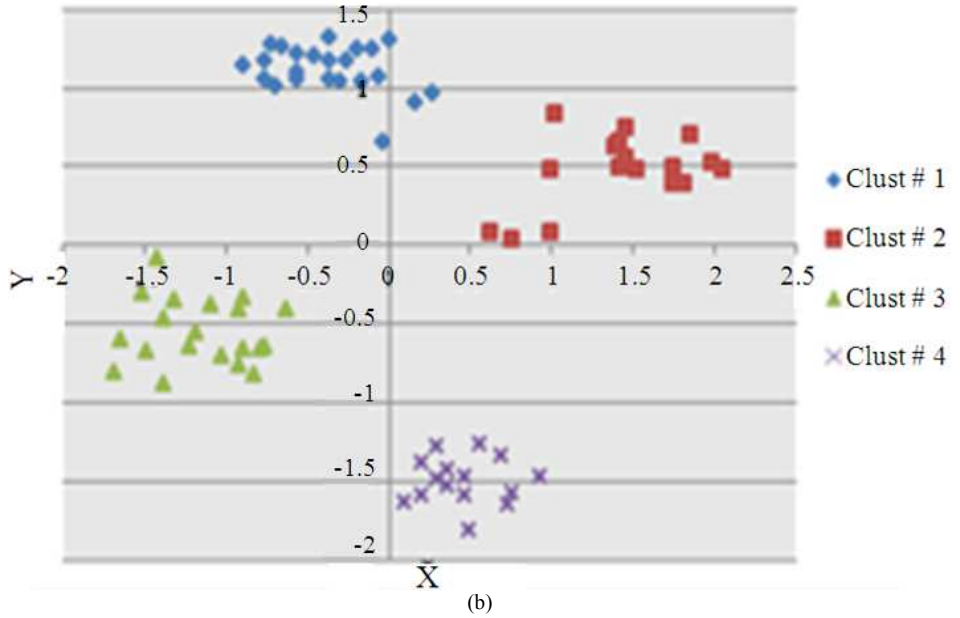
	F-measure		Rand index	
Datasets	μ	σ	μ	σ
Wine	0.8384	0.0503	0.8402	0.0288
Vertebral Column_2C	0.7015	0.0127	0.5619	0.0046

Table 6. Comparison of WDBC and Ruspini by intra measures

Datasets	----------Variance----------		----------Dunn index----------	
DBC	0.8505	0.0698	11.9346	2.8616
Ruspini	0.3782	0.0091	9.6725	4.8174

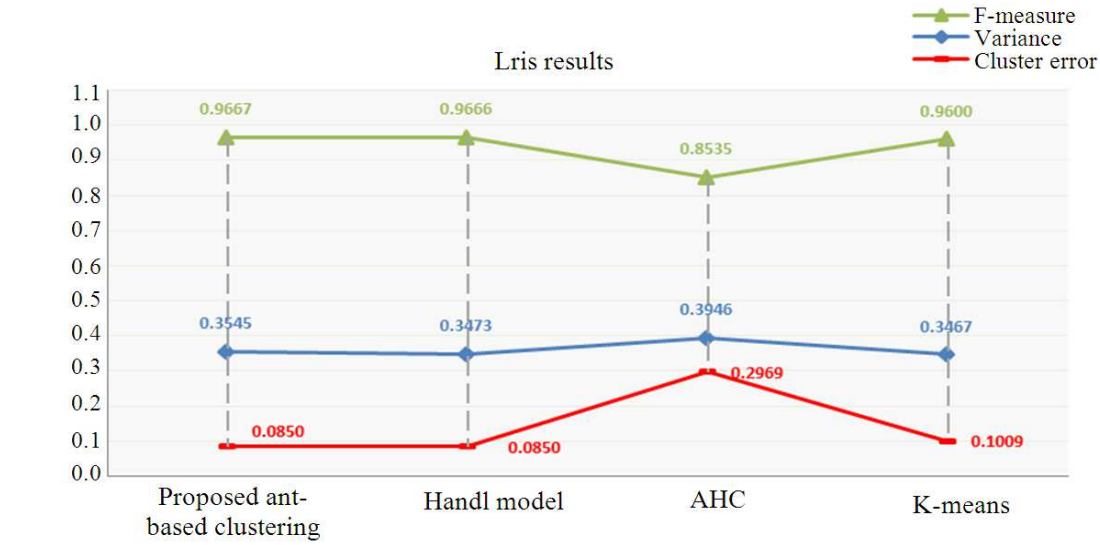


(a)

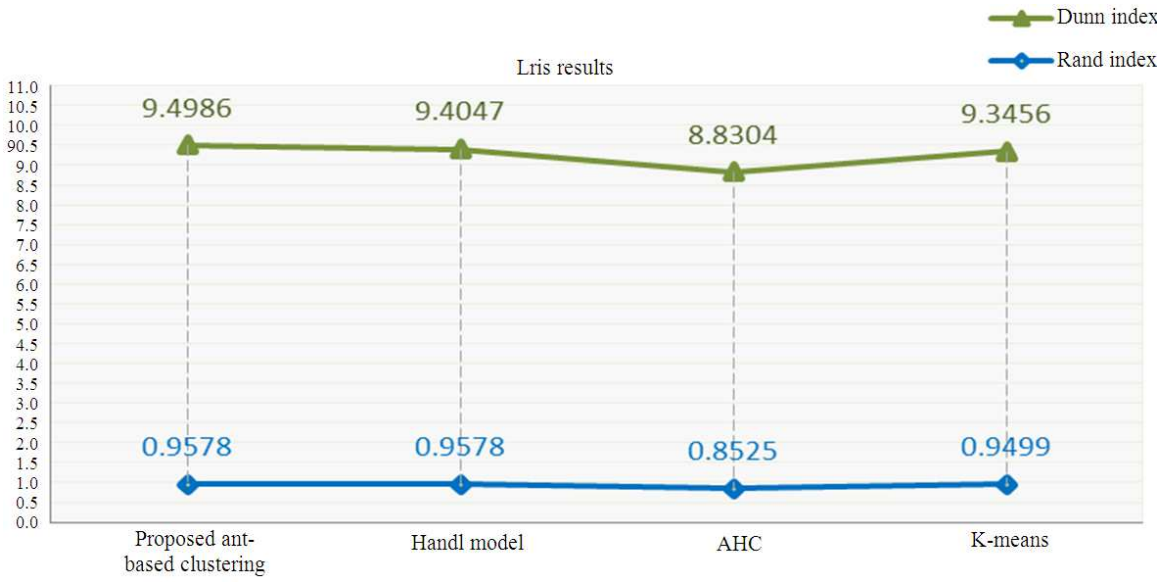


(b)

Fig. 2. Data objects distribution according to their best Variance and Dunn Index values. (a) WDBC and (b) Ruspini



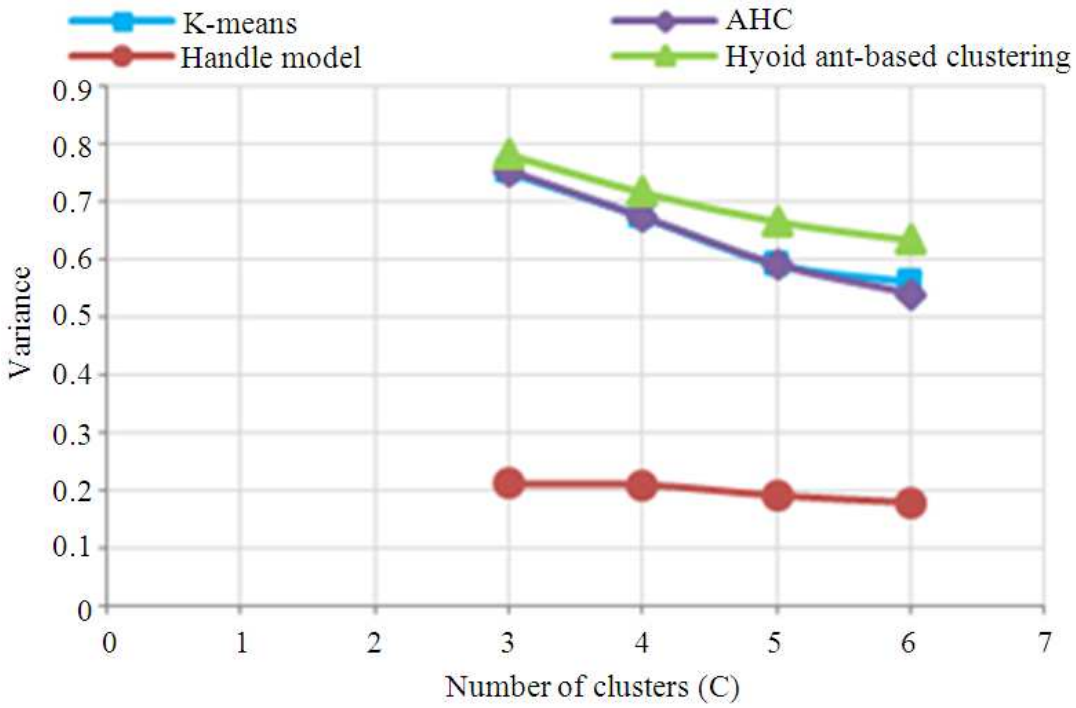
(a)



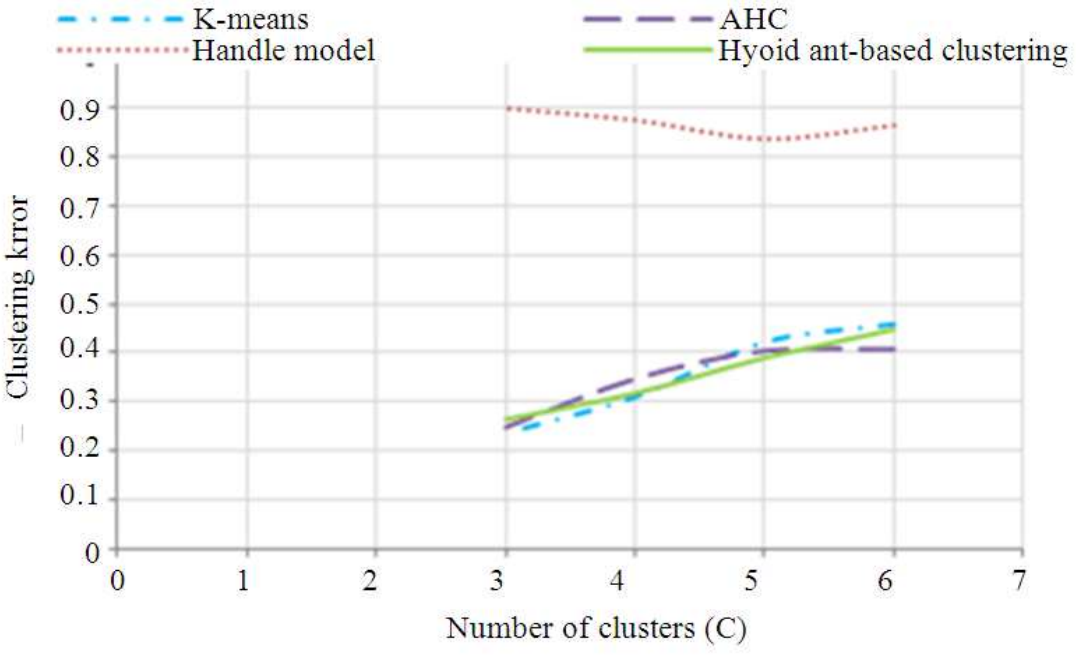
(b)

Fig. 3. Findings for Iris data set by the hybrid Ant-based clustering, Handl Model, (AHC) and K-means algorithms. (a) F-measure and Variance and (b) Dunn Index and Rand Index

The best Dunn index value obtained for the WDBC dataset by the proposed ant-based clustering was 13.83527. This value was obtained in 64% of the total number of runs for correct cluster numbers, which for the WDBC dataset was 2 (**Table 7**). The Handl model, AHC and k-means, on the other hand, failed to attain the value in any of their runs. For the WDBC dataset, the proposed ant-based clustering was able to obtain the lowest clustering error value (0.29184) corresponding to two clusters and a value of 0.3469 corresponding to three-cluster groups; while the other algorithms recorded clustering errors near to a value of one such as the Handl model and thus this model generated too many clusters in most of the runs.



(a)



(b)

Fig. 4. Average results for Wine data set: (a) Variance Vs. Number of Clusters (C) and (b) Clustering Error Vs. Number of Clusters (C)

Table 7. Comparison of Performance

Datasets	Measures	Hybrid ant-based clustering	Handl model	AHC	K-means
Iris	f-Measure	0.9667	0.9666	0.85350	0.9600
	Rand Index	0.9578	0.9578	0.85250	0.9499
	Variance	0.3545	0.3473	0.39460	0.3467
	Dunn Index	9.4986	9.4047	8.83040	9.3456
	Cluster Error	0.0850	0.0850	0.29690	0.1009
Wine	f-Measure	0.9079	0.3470	0.89650	0.9031
	Rand Index	0.8880	0.5530	0.87770	0.8822
	Variance	0.7714	0.2414	0.75570	0.7492
	Dunn Index	10.1659	25.0638	10.37470	9.8713
	Cluster Error	0.2252	0.8990	0.24610	0.2369
WDBC	f-Measure	0.9186	0.6855	0.59230	0.6816
	Rand Index	0.8543	0.5352	0.52410	0.5675
	Variance	0.8632	1.4135	1.83950	1.4683
	Dunn Index	14.0531	3.9224	9.19810	8.6547
	Cluster Error	0.2918	0.9312	0.95340	0.8666
Vertebral Column_2C	f-Measure	0.6726	0.7029	0.60670	0.6623
	Rand Index	0.5541	0.5676	0.50670	0.5500
	Variance	1.1212	1.4078	1.19420	1.1032
	Dunn Index	15.4196	5.1401	19.48130	19.9776
	Cluster Error	0.8946	0.8676	0.98970	0.9030
Ruspini	f-Measure	1.0000	0.4662	1.00000	1.0000
	Rand Index	1.0000	0.6626	1.00000	1.0000
	Variance	0.3740	0.6313	0.36920	0.3692
	Dunn Index	12.7141	1.1407	12.48740	12.4874
	Cluster Error	0.0000	0.6840	0.00000	0.0000

The best results for Column_2C data are shown in **Table 7**; the f-measure, Rand index, variance, Dunn index and clustering error values for the proposed ant-based clustering algorithm were 0.6726, 0.5541, 1.1212, 15.4196 and 0.8946 respectively. Other algorithms attained slightly better results, such as those with the Handl model, where the f-measure was 0.7029; but there were inferior results for k-means and AHC: their f-measure values are 0.6067 and 0.6623.

For Ruspini, a small data set composed of 75 data objects with four clusters, the proposed ant-based clustering, k-means and AHC all attain the same optimal results. Their f-measure is one, clustering error is zero and variance is 0.3740 (**Table 7**). The Handl model, however, attains inferior results since its f-measure value is 0.4662, has a variance value of 0.6313 and clustering error value of 0.6840.

6. CONCLUSION

In this thesis, a hybrid ant-based clustering algorithm is proposed to improve ants' decisions, picking up and dropping off data objects with useful information collected from their environment to contribute to solving cluster problems of assigning scattered data objects to homogeneous clusters. The hybrid ant-based clustering algorithm is inspired from the ACC, AHC and DBSCAN algorithms. The hybrid algorithm has been tested on several real standard datasets. Experimental results showed that the hybrid ant-based clustering method is comparable to the other clustering algorithms in terms of validity measure. Moreover, the method has achieved a higher degree of clustering accuracy for some datasets. The overall results indicate that the hybrid ant-based clustering algorithm is functional as a heuristic clustering algorithm.

In future work, it would be interesting to investigate the behaviour of the ant-based clustering algorithm using other types of heuristic clustering methods. In addition, there is a need to study adaptive threshold strategies and to operate the validity measures as a viable tool to deliver useful local and global information about the cluster environment to enhance the performance of the ant-based clustering algorithm.

7. REFERENCES

Beekman, M., G.A. Sword and S.J. Simpson, 2008. Biological Foundations of Swarm Intelligence. In: Swarm Intelligence: Introduction and Applications, Blum, C. and D. Merkle, (Eds.), Springer-Verlag, Berlin Heidelberg, ISBN-10: 3540740899, pp: 3-41.

Blum, C. and X. Li, 2008. Swarm Intelligence in Optimization. In: Swarm Intelligence: Introduction and Applications, Blum, C. and D. Merkle, (Eds.), Springer-Verlag, Berlin Heidelberg, ISBN-10: 3540740899, pp: 43-85.

Boryczka, U., 2008. Ant clustering algorithm. Intell. Inform. Syst., 1: 377-386.

Cai, F., N.A. Le-Khac and M.T. Kechadi, 2012. Clustering approaches for financial dataanalysis: A survey. Proceedings of the 8th International Conference on Data Mining, (DM' 12), Las Vegas, Nevada, USA., pp: 105-111.

Deneubourg, J.L., S. Goss, N. Franks, A. Sendova-Franks and C. Detrain *et al*., 1991. The dynamics of collective sorting: Robot-like ants and ant-like robots. Proceedings of the 1st International Conference on Simulation of Adaptive Behavior on From Animals to Animats, (AA' 91), Citeseerx, pp: 356-363.

Dhiraj, K., 2009. Study on clustering techniques and application to microarray gene expression bioinformatics data. MSc., Thesis, National Institute of Technology, India.

Dorigo, M. and T. Stutzle, 2004. Ant Colony Optimization. 1st Edn., MIT Press, Cambridge, Mass., ISBN-10: 0262042193, pp: 305.

Elavarasi, S.A., J. Akilandeswari and B. Sathiyabhama, 2011. A survey on partition clustering algorithms. Int. J. Enterprise Comput. Bus. Syst., 1: 1-14.

Engelbrecht, A.P., 2007. Computational intelligence: An introduction. 2nd Edn., John Wiley and Sons, Chichester, ISBN-10: 0470512504, pp: 628.

Hameurlaine, M., A. Moussaoui and H. Cherroun, 2012. AntMeans: A new hybrid algorithm based on ant colonies for complex data mining. Int. J. Comput. Appli., 60: 6-12. DOI: 10.5120/9782-4314

Handl, J., J. Knowles and M. Dorigo, 2006. Ant-based clustering and topographic mapping. Artif. Life, 12: 35-61. PMID: 16393450

Hastie, T.J., R.J. Tibshirani and J. Friedman, 2009. The Elements of Statistical Learning: Data Mining, Inference and Prediction. 2nd Edn., Springer, New York, ISBN-10: 0387848584, pp: 745.

Inkaya, T., 2011. A methodology of swarm intelligence application in clustering based on neighborhood construction. The Graduate School of Natural and Applied Sciences of Middle East Technical University.

Jafar, O.A.M. and R. Sivakumar, 2010. Ant-based clustering algorithms: A brief survey. Int. J. Compu. Theory Eng., 2: 787-796. DOI: 10.7763/IJCTE.2010.V2.242

Jain, A.K. and S. Maheswari, 2012. Survey of recent clustering techniques in data mining. Int. J. Comput. Sci. Manage. Res., 1: 72-78.

Lumer, E.D. and B. Faieta, 1994. Diversity and adaptation in populations of clustering ants. Proceedings of the 3rd International Conference on Simulation of Adaptive Behavior: From Animals to Animats 3, (SABAA' 84), MIT Press, Cambridge, MA, USA., pp: 501-508.

Martens, D., B. Baesens and T. Fawcett, 2011. Editorial survey: Swarm intelligence for data mining. Mach. Learn., 82: 1-42. DOI: 10.1007/s10994-010-5216-5

Mooi, E.A. and M. Sarstedt, 2011. A Concise Guide to Market Research: The Process, Data and Methods Using IBM SPSS Statistics. 1st Edn., Springer, Berlin, ISBN-10: 3642125417, pp: 307.

Nazeer, K.A., M. Sebastian and S.M. Kumar, 2013. A novel harmony search-K means hybrid algorithm for clustering gene expression data. Bioinformation, 9: 84-88. DOI: 10.6026/97320630009084

Parimala, M., D. Lopez and N.C. Senthilkumar, 2011. A survey on density based clustering algorithms for mining large spatial databases. Int. J. Adv. Sci. Technol., 31: 59-66.

Ruspini, E.H., 1970. Numerical methods for fuzzy clustering. Inform. Sci., 2: 319-350. DOI: 10.1016/S0020-0255(70)80056-1

Tan, P.N., M. Steinbach and V. Kumar, 2006. Introduction to data mining. Boston. University of Minnesota.

Villwock, R. and M.T.A. Steiner, 2011. Performance analysis of a proposed ant-based clustering algorithm. Iberoamerican J. Ind. Eng., 3: 184-197.

Weili, Z., 2009. An improved entropy-based ant clustering algorithm. Proceedings of the WASE International Conference on Information Engineering, Jul. 10-11, Taiyuan, Shanxi, China, pp: 41-44. DOI: 10.1109/ICIE.2009.157

Zaharie, D. and F. Zamfirache, 2005. Dealing with noise in ant-based clustering. Proceedings of the IEEE Congress of Evolutionary Computation, Sept. 2-5, IEEE Xplore Press, pp: 2395-2401. DOI: 10.1109/CEC.2005.1554993

Zhe, G., L. Dan, A. Baoyu, O. Yangxi and C. Wei *et al*., 2011. An analysis of ant colony clustering methods: models, algorithms and applications. Int. J. Adv. Comput. Technol., 3: 112-121.

Frank, A. and A. Asuncion, 2010. UCI Machine learning repository. University of California.