INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

Information Sciences 178 (2008) 2816–2831

Aggregation pheromone density based data clustering

Ashish Ghosh [a,*], Anindya Halder [b], Megha Kothari [c], Susmita Ghosh [c]

[a] *Machine Intelligence Unit and Center for Soft Computing Research, Indian Statistical Institute, Kolkata, India*
[b] *Center for Soft Computing Research, Indian Statistical Institute, Kolkata, India*
[c] *Department of Computer Science and Engineering, Jadavpur University, Kolkata, India*

Received 17 May 2007; received in revised form 3 December 2007; accepted 25 February 2008

Abstract

Ants, bees and other social insects deposit pheromone (a type of chemical) in order to communicate between the members of their community. Pheromone, that causes clumping or clustering behavior in a species and brings individuals into a closer proximity, is called aggregation pheromone. This article presents a new algorithm (called, APC) for clustering data sets based on this property of aggregation pheromone found in ants. An ant is placed at each location of a data point, and the ants are allowed to move in the search space to find points with higher pheromone density. The movement of an ant is governed by the amount of pheromone deposited at different points of the search space. More the deposited pheromone, more is the aggregation of ants. This leads to the formation of homogenous groups of data. The proposed algorithm is evaluated on a number of well-known benchmark data sets using different cluster validity measures. Results are compared with those obtained using two popular standard clustering techniques namely average linkage agglomerative and *k*-means clustering algorithm and with an ant-based method called adaptive time-dependent transporter ants for clustering (ATTA-C). Experimental results justify the potentiality of the proposed APC algorithm both in terms of the solution (clustering) quality as well as execution time compared to other algorithms for a large number of data sets.

Keywords: Aggregation pheromone; Ant colony optimization; Swarm intelligence; Data clustering

1. Introduction and motivation

In the literature a wide variety of clustering algorithms have been proposed for different applications [18]. The fundamental problem of clustering is to partition a given data set into groups, such that the data points in a cluster are more similar to each other than points in different clusters [18]. In clustering process there are no predefined classes and training data patterns that would show what kind of desirable relations should be valid among the data; this feature distinguishes clustering from classification [18]. Many clustering methods are available in literature, which can be broadly classified into following types [18] (i) partitional clustering, (ii) hierarchical clustering, (iii) density based clustering, (iv) graph based clustering, and (v) model based cluster-

* Corresponding author.
E-mail address: ash@isical.ac.in (A. Ghosh).

ing. In this article, an aggregation pheromone density based clustering algorithm is proposed which is inspired by the aggregation behavior found in ants and other social insects.

The social insects' behavior such as finding the best food source, building of optimal nest structure, brooding, protecting the larva, guarding, etc. show intelligent behavior on the swarm level [9]. A swarm behavior is not determined just by the behavior of individuals, but the interactions among individuals play a vital role in shaping the swarm behavior [9]. Computational modeling of swarms' behavior is found to be useful in various application domains like, function optimization [38,39], finding optimal routes [6], scheduling [8], image and data analysis [42]. Different applications originated from the study of different types of swarms. Among them, most popular ones are ant colonies [18,39] and bird flocks [9]. ant colony optimization (ACO) [7] and aggregation pheromone systems (APS) [38,39] are computational algorithms modeled on the behavior of the ant colonies. ACO [7] algorithms are designed to emulate ants' behavior of laying pheromone on the ground while moving to solve optimization problems. Pheromone is a type of chemical emitted by an organism to communicate between members of the same species. Pheromone, which is responsible for clumping or clustering behavior in a species and brings individuals into closer proximity, is known as aggregation pheromone. Thus, aggregation pheromone causes individuals to aggregate around good positions which in turn produce more pheromone to attract individuals of the same species. In APS [38,39], a variant of ACO, this behavior of ants is used to solve real parameter optimization problems. A model for solving continuous optimization problems [35] was also proposed as an extension of ant colony optimization (ACO) problem.

In the present article an aggregation pheromone based algorithm (APC) is proposed for data clustering. In order to show the effectiveness of the proposed algorithm we have considered 11 benchmark data sets and four performance (cluster validity) measures. Results are compared with two standard popular clustering algorithms (namely, average linkage agglomerative hierarchical clustering and k-means) and with an ant-based clustering, ATTA-C [15]. Experimental results justify the potentiality of the proposed APC method both in terms of the clustering quality as well as execution time for most of the data sets.

The article is organized in seven sections. Section 2 gives a short survey of related research. Section 3 describes the proposed method. Other chosen methods, used for comparison, are briefly described in Section 4. Section 5 describes the performance (cluster validity) measures. Experimental results are provided in Section 6. Finally, Section 7 concludes the paper.

2. Related work

Numerous abilities of ants have inspired researchers for designing various clustering techniques [5,23]. Several species of ants cluster their corpses into "cemeteries" in an effort to clean up their nests. Experimental work illustrates that ants group corpses, which are initially randomly distributed in space, into clusters, within a few hours. It seems that some feedback mechanism (using local density or similarity of data items) determines the probability that an ant will pick up or drop a corpse. Such behavior is used as a model to design algorithms for clustering data [3].

The earliest model using this concept was proposed by Deneubourg et al. [5] where a population of ants, moving randomly on a grid, was allowed to pick up or drop corpses (data points) so as to cluster them.

Lumer and Faieta [23] further generalized this method and proposed an algorithm known as *Ant Colony Clustering*, which was applied for exploratory data analysis. In both of these works, data movements were implemented through the ants' movement requiring extra information storage and computational burden as the ants make idle movement while carrying no data object. Moreover, in this model, ants carrying isolated corpses (data items) make everlasting move since they never find a proper location to drop them. This consumes a large amount of computation time.

To speed up convergence and to reduce parameter settings, Monmarché et al. [25,26] proposed an interesting hybridization of this algorithm with *k-means* [37] algorithm, and named it as *AntClass*. They compared it with traditional *k-means* [37] and *ISODATA* [1] clustering algorithms on various data sets using classification error as evaluation criterion. Although *AntClass* algorithm gives satisfactory results, the computation time is high. In *AntClass* algorithm, objects (data points) are scattered randomly on the grid board. As a result, the objects in a high density region may be dispersed in different cells and it may need longer time for the ants to collect similar objects into one cell.

In another attempt to speed up this process, Liu et al. [22] proposed *DBAntCluster* algorithm, in which the data distribution property is incorporated into the *Ant Colony Clustering* algorithm. Here, first the high density clusters are found by *DBSCAN* [10] algorithm and the clusters so formed are scattered on the grid board as a special kind of objects together with the single objects in the data set. Afterwards, *Ant Colony Clustering* algorithm is used to cluster the data objects on the grid board.

To enable an unbiased interpretation of the solutions obtained using ant based clustering algorithms, Handl et al. [14,15] proposed a method to determine suitable parameter settings across different test sets. They also suggested a technique to convert the spatial embedding generated by the ant algorithms, which implicitly contains clusters, to an explicit partitioning of the data set. They used different analytical measures to evaluate the results, on synthetic and real data sets, obtained by k-means [24], agglomerative average linkage clustering [41] and one dimensional self organizing map [19], and showed that the ant-based algorithms perform well.

Ramos et al. [30,31] developed an ant clustering system called *ACLUSTER*, for textual document clustering and retrieval of digital images. Unlike *Ant Colony Clustering* algorithm as developed by Lumer and Faieta [23], here ants do not move randomly, rather they move according to some transition probabilities depending on the spatial distribution of the pheromone across the environment. This eliminates the need of short term memory which was required earlier in Lumer and Faieta model [23] for storing past movements of ants. They used the combination of following two independent response threshold functions, associated with different environmental conditions (i) number of objects in an area, and (ii) their similarity.

To improve performance, stability and convergence of the *Ant Colony Clustering* algorithm of Lumer and Faieta [23], Vizine et al. [40] proposed *An Adaptive Ant Clustering Algorithm* with (i) a progressive vision field that allows ants to see over a wider area, (ii) pheromone heuristics to promote reinforcement for dropping of objects at more dense regions of the grid, and (iii) cooling schedule of the parameters that controls the probability of ants picking up objects from the grid. They evaluated their algorithm on a number of well-known benchmark data sets as well as on a real world bioinformatics data set. The modified model is found to have significant improvement over the results obtained with *Ant Colony Clustering* algorithm.

Another approach for ant colony based clustering is proposed by Runkler [32]. The paper shows how objective function based clustering models such as hard and fuzzy c-means can be optimized using a particular extension of simplified ant optimization algorithm. Candidate solutions (in original ant system) that violate any acceptability condition are stored in a tabu list and are then excluded from the solution generation. The contributions of the algorithm are (i) a simplified version of ACO algorithm, (ii) a fuzzification of ACO algorithm and (iii) an application of both the algorithms to the minimization of hard and fuzzy clustering models.

The ACO algorithm for data clustering proposed by Shelokar et al. [33] states that a set of concurrent distributed agents collectively discovers a sensible organization of objects for a given data set and each agent discovers a possible partition of objects in a given data set. The level of partitioning is measured subject to some (Euclidean distance) metric. Information associated with an agent about clustering of objects is accumulated in the global information hub (pheromone trail matrix) and is used by the other agents to construct possible clustering solutions and improve them iteratively.

Zhang et al. [43] applied an ant colony optimization algorithm to cluster analysis. Ant colony algorithm has many advantages but it has the shortcomings of getting stuck at local optima. To overcome this problem they applied particle swarm optimization and proposed an improved clustering algorithm which avoids early convergence.

In a very recent work Sinha et al. [34] proposed ant colony based hybrid optimization for the data clustering. ACO technique along with simulated annealing, tournament selection, tabu search and density distribution are used to solve clustering problem. Tournament selection is used to find the fittest path for the ants to visit. Tabu search restricts the movement of artificial ants to avoid using the same path again and again.

A very recent comprehensive review on ant-based and swarm-based clustering is done by Handel and Meyer [17].

Most of the ant-based clustering algorithms, developed till now, are inspired by the ants' property of piling up the corpses to clean the nest. Besides nest cleaning, many functions of aggregation behavior have been observed in ants and ant like agents [2,28,36]. These include foraging-site marking and mating, finding shelter and defense. For example, after finding safe shelter, cockroaches produce a specific pheromone with their

excrement, which attracts other members of their species [36]. Based on the similar property i.e., ants need to find comfortable and secure environment to sleep, Chen et al. [4] proposed *Ant Sleeping Model* which makes ants to group with those that have similar physiques. They defined a fitness function to measure the ants' similarity with their neighbors. They stated that when an ant's fitness is low, it has a higher probability to wake up and stay in active state. Thus an ant will leave its original position to search for a more secure and comfortable position to sleep. Since each individual ant uses only a little local information to decide whether to be in active state or sleeping state, the whole ant group dynamically self organizes into distinctive, independent subgroups. Using similar concept Tsutsui et al. [38,39] used *Aggregation Pheromone Systems* for continuous function optimization where aggregation pheromone density is defined by a density function in the search space.

In an interesting work Gutjahr [12] showed that for a particular ACO algorithm its current solutions converge to an optimal solution with probability exactly one.

As mentioned above, many functions of aggregation behavior have been observed in ants and ant like agents. Inspired by this behavior found in ants and other similar agents, in our earlier work preliminary attempts are made for solving clustering [20], image segmentation [11], and change detraction [21] problems with encouraging results.

3. Proposed methodology

As mentioned in the introduction, aggregation pheromone brings individuals into closer proximity. This group formation nature of aggregation pheromone is being used as the basic idea of the proposed technique. Here each ant represents one data. The ants move with an aim to create homogenous groups. The amount of movement of an ant towards a point is governed by the intensity of aggregation pheromone deposited by all other ants at that point. This gradual movement of ants in due course of time results in formation of groups or clusters. Fig. 1 depicts the block diagram of the proposed aggregation pheromone density based clustering (APC) method.

The first step of the proposed methodology aims to form clusters using aggregation behavior of ants. At each location of a data point, an ant is placed. The amount of pheromone deposited by an ant at a particular

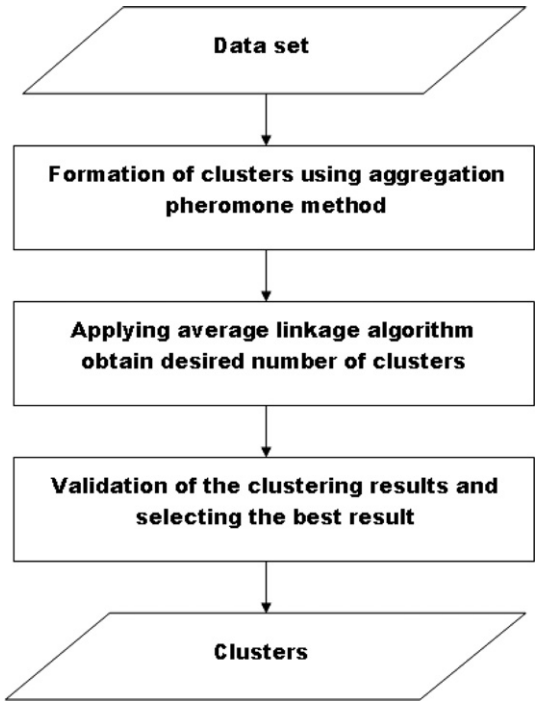

Fig. 1. Block diagram of the proposed scheme.

location depends on its distance from it. Less is the distance, more is the deposition of pheromone. Thus aggregation pheromone density at a particular location depends on the number of ants in its closer proximity. More is the number of ants in its closer proximity, the higher is the aggregation pheromone density. The ants are allowed to move in the search space to find out the points with higher pheromone density. The movement of an ant is governed by the amount of pheromone deposited at different points of the search space. More the deposited pheromone, more is the aggregation of ants. This leads to the formation of homogenous groups or clusters. The number of clusters so formed might be more than the desired number of clusters. So as to obtain the desired number of clusters, in the second step *agglomerative average linkage* algorithm [41] is applied. Finally, the clustering results obtained are validated by using cluster validity measures. In this article four different cluster validity measures are used namely, Rand [37], Jacard [37], S_dbw [13] and Beta [29].

3.1. Aggregation pheromone density based clustering

Consider a data set of n patterns $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \ldots, \boldsymbol{x}_n$ and a population of n-ants $a_1, a_2, a_3, \ldots, a_n$ where an ant a_i represents the data pattern $\boldsymbol{x}_i$. Each individual ant emits pheromone in its neighborhood. The intensity of pheromone emitted by an individual ant a_i (located at $\boldsymbol{x}_i$) decreases with its distance from $\boldsymbol{x}_i$. Thus the pheromone intensity at a point closer to $\boldsymbol{x}_i$ is more than those at other points that are farther from it. To achieve this, the pheromone intensity emitted by ant a_i is modeled by a Gaussian distribution. The pheromone intensity deposited at $\boldsymbol{x}$ by an ant a_i (located at $\boldsymbol{x}_i$) is thus computed as

$$\Delta\tau(a_i, \boldsymbol{x}) = \exp^{-\frac{d(\boldsymbol{x}_i, \boldsymbol{x})^2}{2\delta^2}} \tag{1}$$

where δ denotes the spread of Gaussian function and $d(\boldsymbol{x}_i, \boldsymbol{x})$ is the Euclidean distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}$. Total aggregation pheromone density at $\boldsymbol{x}$ deposited by the entire population of n ants is computed using Eq. (2)

$$\Delta\tau(\boldsymbol{x}) = \sum_{i=1}^{n} \exp^{-\frac{d(\boldsymbol{x}_i, \boldsymbol{x})^2}{2\delta^2}}. \tag{2}$$

Now, an ant a_i which was initially at location $\boldsymbol{x}_i$ moves to the new location $\boldsymbol{x}'_i$ (computed using Eq. (3)) if the total aggregation pheromone density at $\boldsymbol{x}'_i$ is greater than that at $\boldsymbol{x}_i$. The movement of an ant is governed by the amount of pheromone deposited at different points in the search space; and is defined as

$$\boldsymbol{x}'_i = \boldsymbol{x}_i + \eta \cdot \frac{\mathrm{Next}(a_i)}{n} \tag{3}$$

where

$$\mathrm{Next}(a_i) = \sum_{j=1}^{n} (\boldsymbol{x}_j - \boldsymbol{x}_i) \cdot \exp^{-\frac{d(\boldsymbol{x}_j, \boldsymbol{x}_i)^2}{2\delta^2}} \tag{4}$$

with η (a proportionality constant) as the step size. This process of finding a new location continues until an ant finds a location where the total aggregation pheromone density is more than its neighboring points. Once the ant a_i finds out such a point $\boldsymbol{x}'_i$, then that point is assumed to be a new potential cluster center, say $\boldsymbol{Z}_j$ ($j = 1, 2, \ldots, C$, C being number of clusters); and the data point with which the ant was associated earlier (i.e., $\boldsymbol{x}_i$) is assigned to the cluster so formed with center $\boldsymbol{Z}_j$. Also the data points that are within a distance of $\delta/2$ from $\boldsymbol{Z}_j$ are assigned to the newly formed cluster. On the other hand, if the distance between $\boldsymbol{x}'_i$ and the existing cluster center $\boldsymbol{Z}_j$ is less than 2δ and the ratio of their densities is greater than *threshold_density* (a predefined parameter), then the data point $\boldsymbol{x}_i$ is allocated to the cluster having cluster center at $\boldsymbol{Z}_j$. Higher value of density ratio represents that the two points are of nearly similar density and hence should belong to the same cluster. The pseudo code of the proposed aggregation pheromone density based clustering (APC) algorithm is given below.

Pseudo Code

Initialize δ, *threshold_density*, η
$C = 0$

for $i = 1 : n$ do
  if (the data pattern $\boldsymbol{x}_i$ is not already assigned to any cluster)
    Compute $\Delta\tau(\boldsymbol{x}_i)$ using Eq. (2).
    label 1:
      Compute new location $\boldsymbol{x}'_i$ using Eq. (3).
      Compute $\Delta\tau(\boldsymbol{x}'_i)$.
    //End of label
    if $(\Delta\tau(\boldsymbol{x}'_i) > \Delta\tau(\boldsymbol{x}_i))$
      Update the location of ant a_i (at $\boldsymbol{x}_i$) to $\boldsymbol{x}'_i$
      and **goto label 1**.
    end of if
    if $(C == 0)$ //If no cluster exists
      Consider $\boldsymbol{x}'_i$ as cluster center $\boldsymbol{Z}_1$ and increase C by one.
      Assign all the data points within a distance of $\delta/2$
      from $\boldsymbol{x}'_i$ to the newly formed cluster.
    else
      for $j = 1 : C$
     if $(\min(\Delta\tau(\boldsymbol{x}'_i), \Delta\tau(\boldsymbol{Z}_j))/\max(\Delta\tau(\boldsymbol{x}'_i), \Delta\tau(\boldsymbol{Z}_j)) > threshold\_density$
     and $d(\boldsymbol{x}'_i, \boldsymbol{Z}_j) < 2\delta)$
          Assign $\boldsymbol{x}'_i$ to $\boldsymbol{Z}_j$.
     else
          Assign $\boldsymbol{x}'_i$ as a new cluster center say, $\boldsymbol{Z}_{C+1}$ and
          increase C by one.
          Assign all the data points that are within a distance of $\delta/2$
          from $\boldsymbol{x}'_i$ to the newly formed cluster $\boldsymbol{Z}_{C+1}$.
       end of if
     end of for
  end of if
  end of if (if the data pattern $\boldsymbol{x}_i \ldots$)
end of for

3.2. Merging of clusters

In the proposed method (described above), we have applied the APC algorithm on the whole data set in only one pass. The number of clusters produced (depending on the parameter values) may be more than the desired number of clusters. To obtain the desired number of clusters, we applied the average linkage agglomerative hierarchical clustering algorithm (average linkage in short) [41] for merging them. In this sense the two steps are applied in combination (one after another in only one iteration). In other words the algorithm stops after one generation only.

4. Methods compared with

After describing our proposed clustering algorithm APC in the previous section, we are now in a stage to briefly describe the other methods with which the results are compared. We selected two popular conventional clustering techniques (average linkage agglomerative hierarchical clustering and k-means algorithm) and an ant-based method namely, adaptive time-dependent transporter ants for clustering (ATTA-C) [15] for this purpose.

4.1. k-Means

Starting with k randomly-chosen patterns or k randomly defined points inside the hypervolume containing the data set, the k-means algorithm [24] repeatedly (i) (re)assign each pattern (data item) to the closest cluster

center and (ii) (re)computes the current cluster centers (i.e., the average vector of each cluster in data-space). It terminates when no more reassignments of the data points take place. In this way, the intra-cluster variance, that is, the sum of squares of the differences between data items and their associated cluster centers, is locally minimized.

We have used the batch version of the k-means algorithm, that is, cluster centers are recomputed only after reassignment of all data items. k-Means is run repeatedly (20 times) using random initialization of the cluster centers and the average result is listed in Table 2.

4.2. Average linkage agglomerative clustering

As a second method, an agglomerative hierarchical clustering algorithm based on the average linkage metric [41] is used. The algorithm starts with the finest partitioning possible (i.e., singletons) and, in each iteration, merges the two least distant clusters. The distance between two clusters C_i and C_j is computed as the average dissimilarity between all possible pairs of data elements i and j with $i \in C_i$ and $j \in C_j$. The algorithm terminates when the desired number of clusters has been obtained. The algorithm is executed for 20 runs and the average result is reported Table 2.

4.3. Adaptive time-dependent transporter ants for clustering (ATTA-C)

Starting with the basic generic ant algorithm derived from those presented in [5,16,23]. Handl et al. proposed an algorithm called adaptive time-dependent transporter ants for clustering (ATTA-C) [15] to improve and overcome the pitfalls of the previous works such that it can (i) return an explicit partitioning of data by an automatic process, and (ii) work robustly with the same parameter settings across different data sets. As in the earlier works [5,16,23], here also ants are represented as agents that move around the environment, a square grid, in random. Objects (data items) are randomly scattered in this environment and ants can pick up an object, move it and drop it. The probability of picking up and dropping of objects in the ATTA-C algorithm are defined as:

$$P_{\text{pick}}(i) = \begin{cases} 1.0 & \text{if } f(i) \leqslant 1.0, \\ \frac{1}{f(i)^2} & \text{otherwise}, \end{cases} \tag{5}$$

$$P_{\text{drop}}(i) = \begin{cases} 1.0 & \text{if } f(i) \geqslant 1.0, \\ \frac{1}{f(i)^4} & \text{otherwise}, \end{cases} \tag{6}$$

where $f(i)$ is given as

$$f(i) = \max\left(0.0, \frac{1}{\xi^2} \sum_{j \in L} \left(1 - \frac{\text{dis}(i,j)}{\alpha}\right)\right). \tag{7}$$

Here, $\text{dis}(i,j)$ is a dissimilarity function between points in the data-space, ξ^2 is the size of the local neighborhood L, and α is a scaling parameter which determines the percentage of data items on the grid that are classified as similar. Apart from the newly defined 'picking–drooping' probabilities and neighborhood function, few other modifications on 'short term memory with look-ahead', 'radius of perception', 'time varying neighborhood function $f(i)$' and 'cluster retrieval' are made in ATTA-C. Note that ATTA-C has many parameters. The authors have designed some techniques to set them for optimal performance. In our implementation also we have followed the same settings. For more details of the ATTA-C method please see [15].

5. Performance evaluation measures

In order to evaluate the performances of the above described clustering algorithms, in this article we have used four (two external and two internal) cluster validity measures [37]. External cluster validity measures evaluate the results of clustering algorithm based on a pre-specified structure of the data set, i.e. it takes into account the class label information to validate the results. In this article two external validity measures namely

Rand and Jacard [37] are used. Internal cluster validity index, measures the fit between the partition imposed by a clustering algorithm and data itself; class label and other prior information are not used in internal indices. In this article, S_dbw [13] and beta [29] measures are used to validate the clustering results. Following is a description of these validity measures:

- Rand coefficient (R): It determines the degree of similarity between the known correct solution reflecting its class label (group) and the solution obtained by a clustering algorithm [37]. It is defined as

$$R = \frac{\mathrm{SS} + \mathrm{DD}}{\mathrm{SS} + \mathrm{SD} + \mathrm{DS} + \mathrm{DD}}. \tag{8}$$

  SS, SD, DS, DD represent the number of possible pairs of data points i and j where,
  SS: both the data points belong to the same cluster and same group.
  SD: both the data points belong to the same cluster but different groups.
  DS: both the data points belong to different clusters but same group.
  DD: both the data points belong to different clusters and different groups.
  Value of R is in the range $[0,1]$ and *higher the value of R, better is the clustering*.
- Jacard coefficient (J): It is the same as rand coefficient except that it excludes DD and is defined as

$$J = \frac{\mathrm{SS}}{\mathrm{SS} + \mathrm{SD} + \mathrm{DS}}. \tag{9}$$

  Value of J lies in the interval $[0,1]$ and *higher the value of J, better is the clustering*.
- S_dbw: S_dbw index with C number of clusters is based on cluster compactness in terms of intra-cluster variance and inter cluster density [13]. It is defined as

$$\mathrm{S\_dbw}(C) = \mathrm{Scat}(C) + \mathrm{Den}(C); \tag{10}$$

  where $\mathrm{Scat}(C)$ represents the intra-cluster variance and is defined as

$$\mathrm{Scat}(C) = \frac{1}{C}\sum_{i=1}^{C} \|\sigma(\boldsymbol{Z}_i)\|/\sigma(s); \tag{11}$$

  the term $\sigma(s)$ is the variance of the data set and $\sigma(\boldsymbol{Z}_i)$ is the variance of cluster C_i. Inter-cluster density, $\mathrm{Den}(C)$, is defined as

$$\mathrm{Den}(C) = \frac{1}{C-1}\sum_{i=1}^{C}\left(\sum_{j=1, j\neq i}^{C} \frac{\mathrm{den}(\boldsymbol{u}_{ij})}{\max\{\mathrm{den}(\boldsymbol{Z}_i), \mathrm{den}(\boldsymbol{Z}_j)\}}\right); \tag{12}$$

  where $\boldsymbol{Z}_i$ and $\boldsymbol{Z}_j$ are centers of clusters C_i and C_j, respectively and $\boldsymbol{u}_{ij}$ is the mid point of the line segment joining $\boldsymbol{Z}_i$ and $\boldsymbol{Z}_j$. The term $\mathrm{den}(\boldsymbol{u})$ is defined as

$$\mathrm{den}(\boldsymbol{u}) = \sum_{\boldsymbol{x}\in C_i \cup C_j} f(\boldsymbol{x}, \boldsymbol{u}). \tag{13}$$

  The function $f(\boldsymbol{x}, \boldsymbol{u})$ is defined as

$$f(\boldsymbol{x}, \boldsymbol{u}) = \begin{cases} 0 & \text{if } d(\boldsymbol{x}, \boldsymbol{u}) > \mathrm{stdev}, \\ 1 & \text{otherwise}, \end{cases} \tag{14}$$

  where, stdev is the average standard deviation of C clusters and is defined as

$$\mathrm{stdev} = \frac{1}{C}\sqrt{\sum_{i=1}^{C} \|\sigma(\boldsymbol{Z}_i)\|} \tag{15}$$

  and $d(\boldsymbol{x}, \boldsymbol{u})$ is the Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{u}$.*Lower the value of* S_dbw, *better is the clustering*.
- Beta index (β): It computes the ratio of total variation and within class variation [29], and is defined as

$$\beta = \frac{\sum_{i=1}^{C}\sum_{j=1}^{n_i}(\boldsymbol{X}_{ij} - \overline{\boldsymbol{X}})^2}{\sum_{i=1}^{C}\sum_{j=1}^{n_i}(\boldsymbol{X}_{ij} - \overline{\boldsymbol{X}}_i)^2} \tag{16}$$

where $\overline{\boldsymbol{X}}$ is the mean of all the data points and $\overline{\boldsymbol{X}}_i$ is the mean of the data points that belong to cluster C_i, $\boldsymbol{X}_{ij}$ is the jth data point of ith cluster and n_i is the number of data points in cluster C_i. Since the numerator is a constant for a given data set, the value of β is dependent only on the denominator. The denominator decreases with homogeneity in the formed clusters. Therefore, for a given data set, higher the value of β, better is the clustering.

6. Experimental results

Performance of the proposed algorithm has been evaluated using eleven real benchmark data sets taken from the machine learning repository [27].

6.1. Description of data sets

A summery about the data sets is given in Table 1. Wine data has 178 instances of three types of wine with thirteen features. Wisconsin Breast Cancer (WBC) data contains 699 instances distributed in two categories described by nine features of which 16 instances with the missing values are ignored. Sonar data has 208 instances described by 60 attributes distributed in two classes. Thyroid data set has 215 instances of patients with five features describing whether patient is euthyroidism, hypothyroidism and hyperthyroidism (three classes). Glass data set has 214 instances describing six categories of glass on the basis of nine features. Zoo data has 101 instances of animals described by 16 features categorizing animals into seven classes. Lymphography data has 148 instances described by 18 features, distributed in four classes. Balance scale data was generated to model the psychological experimental results. It has 625 instances described by four features, distributed in three classes. Ionosphere is a radar data which consist of 351 instances each with 34 continuous features distributed in two class namely "good" and "bad". This radar data was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power of the order of 6.4 kilowatts. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structures in the ionosphere. "Bad" returns are those that do not; their signals pass through the ionosphere. Image data (training) set has 210 instances drawn randomly from a database of seven outdoor images each having 19 attributes. The images were hand-segmented to create a classification for every pixel. Vowel (deterding data) having 990 instances with 10 features is a dataset for speaker independent recognition of the eleven steady state vowels of British English.

Table 1
Summary of the data sets from machine learning repository

Data set	N	D	C
Wine	178	13	3
WBC	683	9	2
Sonar	208	60	2
Thyroid	215	5	3
Glass	214	9	6
Zoo	101	16	7
Lymphography	148	18	4
Balance scale	625	4	3
Ionosphere	351	34	2
Image	210	19	7
Vowel	990	10	11

N is the total number of data in the data set, D and C represent dimensionality and number of clusters, respectively.

6.2. Role of parameters

The proposed algorithm has three parameters namely η, *threshold_density* and δ.

Here η is the step size. The smaller the step size, more will be the time taken to traverse the search space. The performance of the algorithm in terms of validity measures is found to remain almost constant for a wide range [0.1–1.9] of η. A typical illustration of the variation of the performance measures and execution time (scaled properly) with respect to η is shown in Fig. 2 for wine data set (keeping *threshold_density* fixed at 0.9, and average δ value 0.1625). We have reported results of the experiments with step size $\eta = 1$, as the performance is found to be constant over a wide range around it.

If the ratio of pheromone density of a data point and an already formed cluster center (within 2δ distance) is higher than the *threshold_density* then that point is assigned to the corresponding cluster. This assumes that two close points having nearly similar pheromone density should belong to the same cluster. High *threshold_density* value indicates that pheromone densities of two points (within 2δ) should be very close to come in the same cluster. Less *threshold_density* value indicates that the two close points may reside in the same cluster even if their pheromone densities are not very similar. If the *threshold_density* value is high, it is likely to form large number of clusters in the first phase (before merging the clusters); and if it is less, the number of clusters formed (in the initial phase) may be small. We have executed the algorithm taking different values over the range [0.6–0.9] and on the average 0.9 was found to be a suitable value. For typical illustration of the variation of the performance measurers and execution time with respect to *threshold_density* (keeping $\eta = 1$, and δ value 0.1625) is shown in Fig. 3 for Wine data.

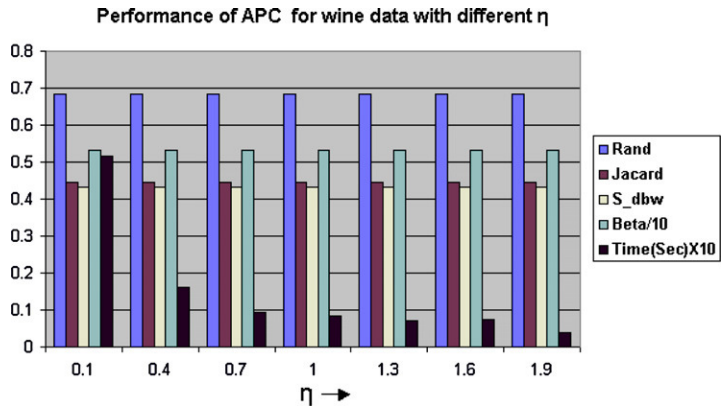

Fig. 2. Performance of APC for wine data with different η.

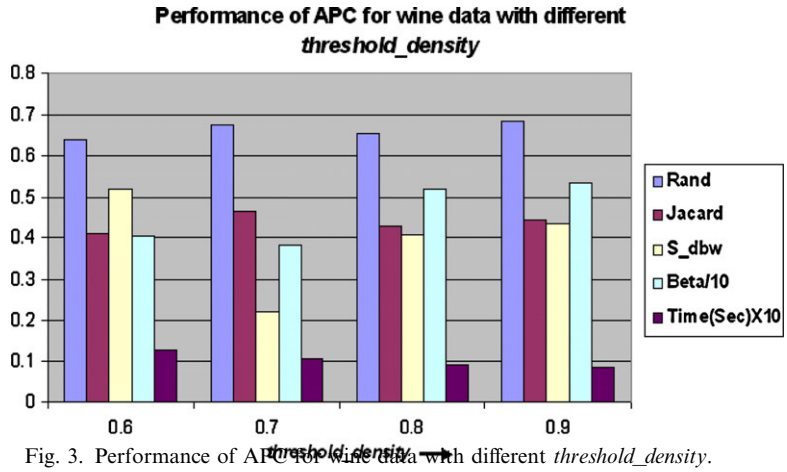

Fig. 3. Performance of APC for wine data with different *threshold_density*.

The algorithm is executed for different δ (spread of the Gaussian) values in the range (0–1]. We determined a stable range of δ for which we got stable compact clusters (i.e. validity measures are stable over this range of δ as shown in Table 2). Stability of the results over that range can also be seen from the smaller values of stan-

Table 2
Validity indices and execution time by the proposed APC, average linkage, *k*-means and ATTA-C algorithms

Data set	Method	Rand	Jacard	S_dbw	Beta	Time
Wine	APC	0.683959 #3 (0.0107)	0.444159 #2 (0.017803)	0.433861 #2 (0.057047)	5.311533 #2 (0.58403)	0.008 #1 (0.002959)
	Average linkage	0.626166 (2.05E−08)	0.424734 (0)	0.473072 #3 (0)	4.508244 #3 (0)	0.05 #3 (0.006666667)
	k-Means	0.716982688 #2 (0.00669725)	0.412723938 #3 (0.00302375)	0.381642813 #1 (0.06943525)	7.37445625#1 (0.1851735)	0.013333333 #2 (0.005163978)
	ATTA-C	0.839787 #1 (0.03464)	0.6311651 #1 (0.03752059)	1.2207669 (0.402096991)	1.0121842 (0.006183323)	7.2 (0.4)
WBC	APC	0.93184 #2 (0.00608)	0.88276 #2 (0.009517)	0.016397 #1 (0.001192)	2.451556 #2 (0.012503)	0.398125 #2 (0.053667886)
	Average linkage	0.892161 (0)	0.823086 (0)	0.017919 #3 (0)	2.406911 #3 (0)	2.4 #3 (0.094074439)
	k-Means	0.925883143 #3 (0.0107)	0.873357 #3 (0.017803)	0.016584286 #2 (0.057047)	2.506990143 #1 (0.58403)	0.027142857 #1 (0.0048795)
	ATTA-C	0.942603 #1 (0.00269844)	0.8996373 #1 (0.00478045)	1.6410922 (0.572984678)	1.0018367 (0.001470227)	13.9 (0.7)
Sonar	APC	0.505435 #1 (0)	0.46113 #1 (6.84E−09)	0.128912 #1 (0)	1.141451 #2 (3E−05)	1.043333333 #3 (0.141503828)
	Average linkage	0.503205 #3 (0)	0.459357 #2 (6.664E−09)	0.146779 #2 (2.36E−09)	1.137581 #3 (0)	0.109310345 #2 (0.010327161)
	k-Means	0.501377889 (0.000765815)	0.335044333 #3 (0.002139166)	0.240741556 #3 (0.058815934)	1.289221222 #1 (0.000878814)	0.041666667 #1 (0.007527727)
	ATTA-C	0.503952 #2 (0.00265762)	0.2679469 (0.030215022)	1.8015625 (0.093667614)	1.0092761 (0.00279416)	7.5 (0.5)
Thyroid	APC	0.65288 #1 (0.00336)	0.597341 #1 (0.002282)	0.047246 #1 (0.002974)	1.479933 #2 (0.008716)	0.002258065 #1 (0.001250237)
	Average linkage	0.59035 (0)	0.559214 #3 (0)	0.050921 #2 (0)	1.415431 #3 (0)	0.074 #3 (0.009660918)
	k-Means	0.649559 #2 (0.013106)	0.58201 #2 (0.013866)	0.1983725 #3 (0.018987)	2.20653 #1 (0.004330127)	0.0125 #2 (0.0051)
	ATTA-C	0.646594 #3 (0.0462011)	0.4105863 (0.08173289)	1.6272899 (0.210468759)	1.0133998 (0.008589679)	6.5 (0.5)
Glass	APC	0.406123 #3 (0.038737)	0.275178 #1 (0.009883)	0.002501 #2 (0.000384)	1.636239 #1 (0.015958)	0.004117647 #1 (0.002072997)
	Average linkage	0.329648 (0)	0.260468 (0)	0.001431 #1 (0)	1.540538 #2 (0)	0.086451613 #3 (0.007549122)
	k-Means	0.704737333 #1 (0.01228682)	0.267611333 #3 (0.029821924)	0.117370833 #3 (0.04787766)	1.523335167 #3 (0.025309007)	0.025 #2 (0.005773503)
	ATTA-C	0.635352 #2 (0.0395196)	0.2699384 #2 (0.035816068)	0.5645105 (0.55729423)	1.009839 (0.004897855)	6.8 (0.4)
Zoo	APC	0.881584 #2 (0)	0.573162 #2 (0)	0.001847 #2 (0)	5.5889 #1 (0)	0.006190476 #1 (0.003400129)
	Average linkage	0.86495 #3 (0)	0.542282 #3 (1.184E−08)	0.001845 #1 (3.78E−11)	5.291049 #2 (0)	0.011875 #3 (0.004031129)
	k-Means	0.79976883 (0.031156935)	0.375798833 (0.044865432)	0.002015 #3 (0.000092)	4.104749 #3 (0.781289623)	0.0096 #2 (0.004383273)
	ATTA-C	0.892775 #1 (0.0231966)	0.6867155 #1 (0.054153267)	0.0096144 (0.000203227)	1.026992 (0.016850339)	6.2 (0.4)

(*continued on next page*)

Table 2 (*continued*)

Data set	Method	Rand	Jacard	S_dbw	Beta	Time
Lymphography	APC	0.614384 #1 (0.002264)	0.369351 #1 (0.000783)	0.003851 #3 (1.03E−06)	1.916803 #3 (0.00087)	0.039230769 #3 (0.008448942)
	Average linkage	0.610682 #2 (0)	0.363923 #3 (1.279E−08)	0.00355 #2 (7.55E−11)	1.998636 #1 (4.74E−08)	0.02826087 #2 (0.003875534)
	k-Means	0.600248167 #3 (0.03257126)	0.333394833 (0.056337114)	0.00351983 #1 (0.000201322)	1.980279333 #2 (0.199353963)	0.02 #1 (0.007071068)
	ATTA	0.578689 (0.0523961)	0.3675606 #2 (0.037358412)	0.879269183 (0.494974512)	1.003955052 (0.00685122)	7.1 (0.538516)
Balance Scale	APC	0.609262 #1 (0)	0.330169 #1 (6.84E−09)	0.901128 #2 (0)	1.393878 #3 (0)	6.012272727 #3 (0.72733651)
	Average linkage	0.579846 #3 (0)	0.297528 #2 (8.374E−09)	1.191576 (0)	1.428571 #1 (0)	2.004285714 #2 (0.091206805)
	k-Means	0.583722875 #2 (0.008168474)	0.296938 #3 (0.008163296)	0.9260325 #3 (0.053084263)	1.4255955 #2 (0.002709547)	0.036 #1 (0.013416408)
	ATTA-C	0.574682 (0.0371669)	0.2248602 (0.074779323)	0.9007734 #1 (0.311370431)	1.0115528 (0.003955764)	15.9 (6.18789)
Ionosphere	APC	0.59009 #1 (1.11E−16)	0.547399 #1 (0)	0.001413 #1 (2.17E−19)	1.311353667 #2 (4.71E−07)	2.251904762 #3 (0.076851743)
	Average linkage	0.54009 #3 (2.27E−16)	0.538399 #3 (0)	0.076896 #2 (2.83E−17)	1.002976 #3 (0)	0.364545455 #2 (0.044798732)
	k-Means	0.587725 #2 (0.001288199)	0.4323255 (0.001365705)	0.619633 #3 (0.004046335)	1.3404645 #1 (1.34E−05)	0.04751 #1 (0.005)
	ATTA-C	0.539776 (1.11E−16)	0.539776 #2 (1.17E−16)	**NA**	**1**	93.9 (3.11288)
Image	APC	0.862411222 #1 (0.003467698)	0.338175778 #1 (5.69E−05)	0.000838556 #1 (0.000135563)	1.692828167 #2 (0.013630157)	0.006086957 #1 (0.00570212)
	Average linkage	0.187924 #3 (0)	0.138458 (0)	0.691172 #3 (2.27E−16)	1.014351 #3 (0.008458221)	0.085217391 #3
	k-Means	0.809896167 #2 (0.019176258)	0.271159833 #2 (0.018531849)	0.2123788 #2 (0.104964638)	4.513256167 #1 (2.003526003)	0.045 #2 (0.005773503)
	ATTA-C	0.177937 (0.105238)	0.1413311 #3 (0.004825319)	**NA**	**1** (0.1851735)	98.6 (4.52106)
Vowel	APC	0.8806112 #1 (0.014916901)	0.23302732 #1 (0.005286738)	0.00036288 #1 (4.73E−06)	2.17061968 #2 (0.041176309)	0.288461538 #1 (0.045667696)
	Average linkage	0.117874 (1.44E−17)	0.089694 (2.87E−17)	0.054676 #3 (1.44E−17)	1.011381 #3 (0)	7.106666667 #3 (0.569275856)
	k-Means	0.865609444 #2 (0.002669295)	0.163830111 #2 (0.003860012)	0.000377444 #2 (3.94E−06)	2.668702444 #1 (0.020342263)	0.478 #2 (0.17207517)
	ATTA-C	0.665422 #3 (0.231287)	0.1180951 #3 (0.01423177)	0.324236556 (0.969693167)	1.004958556 (0.001977096)	11.7 (0.458258)

dard deviation (in Table 2). Average value and standard deviation (shown within bracket) of the chosen range of twenty δ values (for which result of Table 2 is reported) for different data sets is listed in Table 3. If we use single validity measure then δ value (or a range δ values) for which we would get the best result (in terms of that validity measure used) should be taken as the best value. If best result is obtained for a range of δ values then any value of δ from that range or the average of that range should be taken as the parameter value.

6.3. Analysis of results

To find out the effectiveness of the proposed algorithm, experiments were carried out on the previously mentioned eleven benchmark real life data sets. To show the robustness of the algorithms, clustering results are validated using four different cluster validity indices (two external and two internal) as described in Section 5. It is worth mentioning here that any validity measure can be used for this purpose. The results obtained by the proposed APC algorithm are compared with those of average linkage, *k*-means and ATTA-C algorithms. Table 2 gives the average values of different performance indices obtained with different chosen (as mentioned

Table 3
Average value and standard deviation (shown in bracket) for different values of δ for APC method together with number of clusters formed by ATTA-C for different data sets

Data set	Chosen δ for APC	Number of clusters formed by ATTA-C
Wine	0.1625	2.9
	(0.00591608)	(0.3)
WBC	0.284571429	2
	(0.011685767)	(0)
Sonar	0.36	2.9
	(0.006204837)	(0.3)
Thyroid	0.291	3.8
	(0.008514693)	(0.4)
Glass	0.093	3.5
	(0.007937254)	(0.67082)
Zoo	0.185	3.2
	(0.006204837)	(0.4)
Lymphography	0.3295	2.6
	(0.007648529)	(0.663325)
Balance Scale	0.1105	7.9
	(0.006493587)	(2.73679)
Ionosphere	0.256	1
	(0.006055301)	(0)
Image	0.448	1
	(0.00735980)	(0.3)
Vowel	0.1625	5.1
	(0.00591608)	(2.16564)

above) δ values and their corresponding standard deviations (shown in bracket) for each of the data sets obtained by APC. Corresponding results for other algorithm (over 20 runs) like average linkage, k-means and ATTA-C algorithms are also shown in the same table.

The CPU (execution) time, in seconds, needed by the algorithms are also given in the table for comparison. Rank of each algorithm is given depending on its performance index and the execution time (separately) using '#' symbol followed by corresponding rank (from 1 to 3). For example '#1' indicates the best result with respect to either the corresponding performance index or execution time.

All the experiments are performed in a HP Proliant (ML350G4P) terminal with Xeon (3.2 GHz clock speed 800 MHz FSB with one 2MB L2 Cache memory) processor and in Linux environment. Implementation of the algorithms is done in C and C++.

Note that the average number of 'automatically' formed clusters and their standard deviation (shown in bracket) by the ATTA-C algorithm is shown in last column of the Table 3.

It is apparent from Table 2 that in terms of external validity measures (Rand and Jacard index) performance of the proposed APC algorithm is better for most of the data sets (namely Sonar, Thyroid, Lymphography, Balance scale, Ionosphere, Image and Vowel) whereas ATTA-C gives better result for Wine, WBC and Zoo data sets. Only for Glass data set k-means algorithm outperforms other three methods in terms of Rand index; and for Jacard index APC performs the best.

With respect to S_dbw measure the performance of the APC algorithm is the best (compared to other three methods) for six data sets (WBC, Sonar, Thyroid, Ionosphere, Image and Vowel) and 2nd best for Wine, Glass, Zoo and Balance Scale data sets (in these cases performance of the APC is very much comparable to those with the best one). Lower value of S_dbw shows that the clusters formed are compact and well separated. It is to be noted that for Ionosphere and Image data sets, as the number of clusters detected by the ATTA-C method is (refer Table 3) one, S_dbw measure tends to 1/0 and therefore is denoted by NA (not applicable); whereas beta measure becomes 1.

In terms of the beta validity measure, for most of the cases (seven data sets) k-means algorithm outperforms the other three methods. In these cases APC's performance is the 2nd best and for Zoo and Glass data sets it is the best. For Lymphography and Balance Scale data sets, average linkage algorithm performs better com-

pared to other three. In short, APC's performance for most of the data set is very close to the best result in terms of the beta measure.

On an average, for most of the data sets, in terms of all the cluster validity measures, APC either outperforms the other methods or is close to the best results produced by the other algorithms.

Note that from the experimental result it is clear that ATTA-C method fails to automatically detect appropriate number of clusters (see Table 3) for several data sets.

In terms of the execution time proposed APC algorithm performs better for six data sets (Wine, Thyroid, Glass, Zoo, Image and Vowel) compared to other three algorithms, whereas k-means algorithm takes less execution time for five cases (WBC, Sonar, Lymphography, Ionosphere and Balance Scale data sets).

As a whole, experimental results on a large number of real world data sets justify the potentiality of the proposed APC algorithm both in terms of solution (clustering) quality as well as execution time compared to other algorithms.

7. Conclusions

In this article we have proposed a new algorithm for clustering data based on aggregation pheromone density, which is inspired by ants' property to accumulate around points with higher pheromone density. To evaluate the performance of the proposed algorithm we have tested it on 11 benchmark data sets with two external and two internal cluster validity measures. Comparative study of the proposed algorithm with the average linkage agglomerative hierarchical clustering, k-means clustering and an ant-based clustering algorithm (ATTA-C) justifies the potentiality of the proposed methodology. Note that ATTA-C has a large number of parameters to be set. Our earlier work [11,21] on the application of this algorithm shows the usefulness of the proposed method to handle real world problems like image segmentation and change detection of remote sensing images.

Experimental results show that the proposed method performs fairly well both in terms of the solution (clustering) quality and execution time. However, the method has a limitation of specifying the number of clusters. There are wide scope of research for improvement of the proposed algorithm such that it can automatically detect the appropriate number of clusters.

Acknowledgements

Authors would like to acknowledge the Department of Science and Technology, Govt. of India and University of Trento, Italy, the sponsors of the project titled "Advanced Techniques for Remote Sensing Image Processing" jointly being carried out by Jadavpur University and Indian Statistical Institute Kolkata. Also support of the Department of Science and Technology, Govt. of India to the Center for Soft Computing Research through its IRHPA scheme is thankfully acknowledged by Mr. Anindya Halder, Research Scholar, Center for Soft Computing Research, Indian Statistical Institute, Kolkata. Thanks are due to Dr. J. Handl for her constant suggestion and help to implement ATTA-C method. The authors are also thankful to the reviewers for their stimulating comments, which helped to improve the quality of the manuscript.

References

[1] G.H. Ball, D.J. Hall, ISODATA: a novel method of data analysis and pattern classification. Technical report, Stanford Research Institute, Menlo Park, CA, 1965.
[2] W.J. Bell, Chemo-orientation in walking insects, in: W.J. Bell, R.T. Carde (Eds.), Chemical Ecology of Insects, 1984, pp. 93–109.
[3] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence – From Natural to Artificial Systems, Santa Fe Institute in the Sciences of Complexity, Oxford University Press, New York, 1999.
[4] L. Chen, X.H. Xu, Y.X. Chen, An adaptive ant colony clustering algorithm, in: Proceedings of the 3rd International Conference on Machine Learning and Cybernetics, Shanghai, August 2004, pp. 1387–1392.
[5] J.L. Deneubourg, S. Goss, N. Franks, A.S. Franks, C. Detrain, L. Chretien, The dynamics of collective sorting: robot-likes ants and ant-like robots, in: J.A. Meyer, S.W. Wilson (Eds.), Proceedings of the 1st Conference on Simulation of Adaptive behavior: From Animals to Animats, vol. 1, MIT Press/Bradford Books, 1991, pp. 356–365.
[6] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Transaction on Evolutionary Computing 1 (1) (1997) 53–66.

[7] M. Dorigo, V. Maniezzo, A. Colorni, The ant system: optimization by a colony of cooperating agents, IEEE Transaction on SMC-Part B 26 (1) (1996) 29–41.
[8] M. Dorigo, T. Stützle, Ant Colony Optimization, Prentice Hall of India Private Limited, New Delhi, 2005.
[9] A.P. Englebrecht, Computational Intelligence: An Introduction, John Wiley and Sons, 2002.
[10] M. Ester, H.P. Kriegel, J. Sander, X. Xu, A density based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, 1996.
[11] S. Ghosh, M. Kothari, A. Ghosh, Aggregation pheromone density based image segmentation, in: P. Kalra, S. Peleg (Eds.), Proceedings of the 5th Indian Conference on Computer Vision Graphics and Image Processing (ICVGIP'06), LNCS, vol. 4338, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 118–127.
[12] W.J. Gutjahr, ACO algorithms with guaranteed convergence to the optimal solution, Information Processing Letters 82 (3) (2002) 145–153.
[13] M. Halkidi, M. Vazirgiannis, Clustering validity assessment: finding the optimal partitioning of a data set, in: Proceedings of ICDM, 2001, pp. 187–194.
[14] J. Handl, J. Knowles, M. Dorigo, On the performance of ant-based clustering, in: Proceedings of the 3rd International Conference on Hybrid Intelligent Systems, Design and Application of Hybrid Intelligent Systems, IOS press, 2003, pp. 204–213.
[15] J. Handl, J. Knowles, M. Dorigo, Ant-based clustering and topographic mapping, Artificial Life 12 (1) (2006) 35–62.
[16] J. Handl, B. Meyer, Improved ant-based clustering and sorting in a document retrieval interface, in: Proceedings of the 7th International Conference on Parallel Problem Solving from Nature, Lecture Notes in Computer Science, vol. 2439, Springer-Verlag, Berlin, Germany, 2002, p. 913923.
[17] J. Handl, B. Meyer, Ant-based and swarm-based clustering, Swarm Intelligence 1 (2007) 95–113, Published online on 13th November 2007.
[18] A.K. Jain, M.N. Murty, P.J. Flyn, Data clustering: a review, ACM Computing Surveys 31 (3) (1999) 264–323.
[19] T. Kohonen, Self-Organizing Maps, Springer, Berlin, 1997.
[20] M. Kothari, S. Ghosh, A. Ghosh, Aggregation pheromone density based clustering, in: S.P. Mohanty, A. Sahoo (Eds.), Proceedings of the 9th International Conference on Information Technology (ICIT06), IEEE Computer Society Press, Los Alamitos, California, 2006, pp. 259–264.
[21] M. Kothari, S. Ghosh, A. Ghosh, Aggregation pheromone density based change detection in remotely sensed images, in: P. Pal (Ed.), Proceedings of 6th International Conference on Advances in Pattern Recognition (ICAPR'07), World Scientific Publishing Co. Pvt. Ltd., Singapore, 2007, pp. 193–197.
[22] S. Liu, Z.T. Dou, F. Li, Y.L. Huang, A new ant colony clustering algorithm based on DBSCAN, in: Proceedings of the 3rd International Conference on Machine Learning and Cybernetics, Shanghai, August 2004, pp. 1491–1496.
[23] E.D. Lumer, B. Faieta, Diversity and adaptation in populations of clustering ants, in: D. Cliff, P. Husbands, J.A. Meyer, S.W. Wilson (Eds.), Proceedings of the 3rd International Conference on Simulation of Adaptive Behaviour: From Animals to Animats, vol. 3, MIT Press/Badford Books, 1994, pp. 501–508.
[24] L. MacQueen, Some methods for classification and analysis of multivariate observations, Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, vol. 1, University of California Press, CA, 1967, pp. 281–297.
[25] N. Monmarché, On data clustering with artificial ants, in: A.A. Freitas (Ed.), AAAI-99 and GECCO-99 Workshop on Data Mining with Evolutionary Algorithms: Research Directions, Orlando, Florida, 1999, pp. 23–26.
[26] N. Monmarché, M. Slimane, G. Venturini, On improving clustering in numerical database with artificial ants, in: D. Floreano, J.D. Nicoud, F. Mondala (Eds.), Advances in Artificial life, 5th European Conference ECAL'99, Swiss Federal Institute of Technology, Lausanne, Switzerland, September, Lecture Notes in Artificial Intelligence 1974, Springer-Verlag, 1999, pp. 626–635.
[27] D.J. Newman, S. Hettich, C.L. Blake, C.J. Merz, UCI repository of machine learning databases. University of California, Irvine, Department of Information and Computer Sciences, 1998. <http://www.ics.uci.edu/~mlearn/MLRepository.html>.
[28] M. Ono, T. Igarashi, E. Ohno, M. Sasaki, Unusual thermal defence by a honeybee against mass attack by hornets, Nature 377 (6547) (1995) 334–336.
[29] S.K. Pal, A. Ghosh, B. Uma Shankar, Segmentation of remotely sensed images with fuzzy thresholding and quantitative evaluation, International Journal on Remote Sensing 21 (11) (2000) 2269–2300.
[30] V. Ramos, J.J. Merelo, Self-organized stigmergic document maps: Environment as a mechanism for context learning, in: E. Alba, F. Herrera, J.J. Merelo (Eds.), Proceedings of the 1st International Conference on Metaheuristic, Evolutionary and Bio-Inspired Algorithms, Spain, 2002, pp. 284–293.
[31] V. Ramos, F. Muge, P. Pina, Self-organized data and image retrieval as a consequence of inter-dynamic synergistic relationships in artificial ant colonies, in: J. Ruiz del Solar, A. Abraham, M. Koppen (Eds.), Soft-Computing Systems-Design, Management and Applications, Frontiers of Artificial Intelligence and Applications, vol. 87, IOS Press, Amsterdam, 2002, pp. 500–509.
[32] T.A. Runkler, Ant colony optimization of clustering models, International Journal of Intelligence System 20 (12) (2005) 1233–1251.
[33] P.S. Shelokar, V.K. Jayaman, B.D. Kulkarni, An ant colony approach for clustering, Analytica Chimica Acta 509 (2) (2004) 187–195.
[34] A.N. Sinha, N. Das, G. Sahoo, Ant colony based hybrid optimization for data clustering, Journal of Kybernetes 36 (2) (2007) 175–191.
[35] K. Socha, M. Dorigo, Ant colony optimization for continuous domains, European Journal of Operational Research 185 (3) (2008) 1155–1173.
[36] M. Sukama, H. Fukami, Aggregation arrestant pheromone of the German cockroach, *Blattella germanica* (L.) (Dictyoptera: Blattellidae): isolation and structure elucidation of blasttellastanoside-A and B, Journal of Chemical Ecology 19 (1993) 2521–2541.
[37] S. Theodoridis, K. Koutroumbas, Pattern Recognition, third ed., Academic Press, New York, 2006.

[38] S. Tsutsui, Ant colony optimization for continuous domains with aggregation pheromones metaphor, in: Proceedings of the 5th International Conference on Recent Advances in Soft Computing (RASC'04), United Kingdom, December 2004, pp. 207–212.
[39] S. Tsutsui, A. Ghosh, An extension of ant colony optimization for function optimization, in: Proceedings of the 5th Asia Pacific Conference on Simulated Evolution and Learning (SEAL04), Pusan, Korea, 2004.
[40] A.L. Vizine, L.N. de Castro, E.R. Hruschka, R.R. Gudwin, Towards improving clustering ants: an adaptive ant clustering algorithm, Informatica 29 (2005) 143–154.
[41] E. Vorhees, The effectiveness and efficiency of agglomerative hierarchical clustering in document retrieval. Ph.D. Thesis, Department of Computer Science, Cornell University, Ithaca, NY, 1985.
[42] X.N. Wang, Y.J. Feng, Z.R. Feng, Ant colony optimization for image segmentation, in: Proceedings of the 4th International Conference on Machine Learning and Cybernetics, 2005, pp. 5355–5360.
[43] X. Zhang, H. Peng, Q. Zheng, A novel ant colony optimization algorithm for clustering, Proceedings of the 8th International Conference on Signal Processing(ICSP), vol. 3, IEEE Computer Society Press, 2006.