

# Modified ACS Centroid Memory for Data Clustering

<sup>1</sup>Ayad Mohammed Jabbar, <sup>2</sup>Ku Ruhana Ku-Mahamud and <sup>3</sup>Rafid Sagban

<sup>1</sup>Computer Science Department, Shatt Al-Arab University College, Iraq

<sup>2</sup>School of Computing, Universiti Utara Malaysia, Malaysia

<sup>3</sup>Computer Science Department, University of Babylon, Iraq

## Article history

Received: 18-07-2019

Revised: 26-08-2019

Accepted: 11-10-2019

Corresponding Author:

Ayad Mohammed Jabbar

College of Arts and Sciences,

Shatt Al-Arab University,

Basra, Iraq

Email: ayadmohammed@sa-uc.edu.iq

**Abstract:** Ant Colony Optimization (ACO) is a generic algorithm, which has been widely used in different application domains due to its simplicity and adaptiveness to different optimization problems. The key component that governs the search process in this algorithm is the management of its memory model. In contrast to other algorithms, ACO explicitly utilizes an adaptive memory, which is important to its performance in terms of producing optimal results. The algorithm's memory records previous search regions and is fully responsible for transferring the neighborhood of the current structures to the next iteration. Ant Colony Optimization for Clustering (ACOC) is a swarm algorithm inspired from nature to solve clustering issues as optimization problems. However, ACOC defined implicit memory (pheromone matrix) inability to retain previous information on an ant's movements in the pheromone matrix. The problem arises because ACOC is a centroid-label clustering algorithm, in which the relationship between a centroid and instance is unstable. The label of the current centroid value changes from one iteration to another because of changes in centroid label. Thus the pheromone values are lost because they are associated with the label (position) of the centroid. ACOC cannot transfer the current clustering solution to the next iterations due to the history of the search being lost during the algorithm run. This study proposes a new centroid memory (A-ACOC) for data clustering that can retain the information of a previous clustering solution. This is possible because the pheromone is associated with the adaptive instance and not with label of the centroid. Centroids will be identified based on the adaptive instance route. A comparison of the performance of several common clustering algorithms using real-world data sets shows that the accuracy of the proposed algorithm surpasses those of its counterparts.

**Keywords:** Data Clustering, Swarm Intelligence, Optimization Based-Clustering, Ant Colony Optimization

## Introduction

Clustering is an unsupervised learning technique that groups data without any prior information. This approach is an undirected learning technique that can be performed without the need for labeled data. This technique differs from the classification learning technique (Al-Behadili *et al.*, 2018a; Jabbar, 2018; Wahid and Al-Mazini, 2018). Different clustering algorithms aim to classify data as clusters, in which each cluster has a set of members that share similar characteristics and are dissimilar to the features of the members of other clusters (Jabbar *et al.*, 2018). The two approaches that can be used to perform clustering are the

deterministic and stochastic approaches. The deterministic approach includes algorithms that consider clustering as a deterministic clustering problem. In contrast, the stochastic approach includes algorithms that belong to swarm algorithms which consider clustering as an optimization problem. The stochastic approach involves the problem of minimizing or maximizing an objective function to achieve a clustering solution. Both approaches have their limitations but the stochastic approach is more accurate than the deterministic because the target of the former is to optimize the problem, in which the clustering problem is regarded as a complex problem when more than three clusters are present (Shabanzadeh and Yusof,

2015). Ant colony optimization (ACO) was developed to solve different optimization problems, such as a clustering problem, in which the algorithm can produce global clustering solutions. The ACO for clustering (ACOC), which is an optimization algorithm for clustering proposed by Kao and Cheng (2006), entails centroid-based clustering, in which the problem of clustering is considered an assignment problem. ACOC uses the concept of the Ant Colony System (ACS) for clustering problems. The assignment problem is a clustering problem and the algorithm assigns each instance to the nearest centroid based on pheromone value and heuristic information. A major drawback of centroid-based clustering is the label position, in which the label refers to the current cluster number. The algorithm is a dynamic cluster center (centroid) and each iteration is randomly updated. Once updated, the pheromone information is forgotten and the algorithm learning process is lost. In this case, the algorithm is unable to transfer previous information to the next iterations. The algorithm becomes a local search algorithm similar to the K-means algorithm. The pheromone information becomes ineffectual in classification and the algorithm relies on the heuristic only.

This study proposes a new pheromone memory approach that can retain the history of previously recorded information to be used in the learning process, in which such information is a learning tool employed in the algorithm. The new pheromone memory is similar to the one employed in a neural network. In this memory, each centroid is considered an optimization problem that contains  $m$  possibilities with  $n$  attributes, where  $m$  is the number of possibilities in the range of  $[0, 1]$  for each single attribute and  $n$  is the number of attributes for the current centroid. The problem of clustering entails finding the optimal value of every attribute of each centroid. The proposed algorithm presents new construction graphs, in which each ant travels from a single attribute of each centroid. Thus, the optimal clustering solution is produced on the basis of the optimal route between attributes generating a minimum intra-clustering distance. The proposed algorithm relies on pheromone information only and no heuristic information is required. This approach will decentralize the algorithm. The learning process starts with a high exploration rate accepting some worse solutions, but the algorithm eventually moves toward the exploitation of the optimal clustering solution.

The remainder of this paper is organized as follows. Section 2 presents the related research on the ACO-based clustering and highlights the problem of pheromone memory. Section 3 provides the research methodology while Section 4 elaborates on the proposed clustering algorithms. Section 5 explores the experiments and evaluates the proposed algorithm against existing state-of-the-art clustering algorithms. Section 6 details the

discussion of the results. Lastly, Section 7 presents the conclusion of this study and future research.

### Related Research

Clustering is unsupervised learning for organizing similar data in the same cluster and dissimilar ones in another cluster (Ünlü and Xanthopoulos, 2019). The objective is to group data on the basis of a similar characteristic, in which the members of a single cluster are close to one another with suitable distance between clusters (Kumar and Sahoo, 2014). In clustering, all algorithms can produce a clustering solution labeled either as partitional clustering or hierarchical clustering solutions.

The partitional clustering algorithm groups data into flat clusters according to global criteria (Singh *et al.*, 2019). The global criteria minimize the dissimilarity measure within a cluster (intra-clustering variance) and maximizes dissimilarity (inter-clustering variance) between clusters. Similarities between elements within single clusters are higher than the dissimilarities of other elements in other clusters (Nagpal *et al.*, 2013).

The hierarchical clustering algorithm visualizes data as a hierarchical tree that illustrates the fusion or division in each stage, in which the tree contains nested clusters (Popat and Emmanuel, 2014). The hierarchical clustering approach is classified into agglomerative and divisive methods. The agglomerative method starts by merging distinct clusters (objects) based on similarity until a single cluster that contains all members is obtained. The divisive method performs a series of partitions that contain single clusters and successively separates them into multiple sub-clusters (Murtagh, 1983; Zhang and Xia, 2009). K-means is a local search algorithm that is popular in partitional clustering because of its efficiency and simplicity. Nevertheless, this algorithm has a problem in obtaining a global solution because it is easily trapped in local optima, while different centroid initializations produce different clustering solutions (Jain, 2010).

Studies on the K-means algorithm have focused on the optimization approach, which employs metaheuristic algorithms to produce (near) optimal solutions in reasonable time. The clustering problem is formulated as an optimization problem or integrated with local search algorithm such as K-means integrated with a metaheuristic algorithm as a hybrid algorithm to produce more optimal clustering results. There are some methodologies applied for escape from local optima using different local search approaches; these include Variable Neighborhood Search (VNS) and Iterated Local Search (ILS) used in clustering algorithms (Abuhamdah, 2018).

A metaheuristic is a framework of ideas, concepts and operators that is used to solve optimization problems. The metaheuristic algorithm is problem-independent, has different optimization strategies and has been applied successfully to solve the clustering problem. Examples of such applications are the genetic

algorithm (GA), particle swarm optimization (PSO), artificial bee colony (ABC), simulated annealing (SA), firefly algorithm (FA), ant colony optimization (ACO) and hybrid approaches, such as the fuzzy C-means and PSO algorithm (FCPSO) and K-means and PSO algorithm (KPSO) (Al-Behadili, 2018; Binu, 2015) The key component that governs the search method in such algorithms is the memory model. Unlike other approaches, ACO explicitly utilizes an adaptive memory, called the pheromone memory, to retain previous information to adapt the algorithm solution (Stutzle and Linke, 2000).

The first research that uses ACO is by Shelokar *et al.* (2004) who propose an algorithm that relies only on pheromone memory of dimension  $m$  by  $n$ , where  $m$  is the number of clusters and  $n$  is the number of instances. The algorithm starts with an empty solution and each ant travels from one instance to another on the basis of the amount of pheromone between instances and the cluster centroid with the maximum amount of pheromones. However, the algorithm produces premature convergence and infeasible solutions and empty clusters may be produced, particularly in the beginning of the run (Kao and Cheng, 2006).

Various studies have used the concept of dynamic cluster centers to produce clustering results (Kao and Cheng, 2006; Santos *et al.*, 2009; Li and Yao, 2016; David and Kosala, 2018). Cluster centroids are randomly dynamic updates. The algorithms use pheromone and heuristic information as guides for the ant to perform a clustering task. The current pheromone memory is ineffective as ants fail to classify data to the correct group because the amount of pheromone laid by previous ants did not represent the right assignment. This situation occurs because centroids are updated in each iteration and the pheromone entries that have appropriate centroids are not static. This is because the pheromone

value is associated with the centroid and this value will be lost. Thus, the current pheromone memory is a memory merely to retain the pheromone. The algorithm becomes a local search algorithm, which is similar to K-means. In other words, the algorithm lacks the ability to transfer current information about the assignment to the next iteration to be used by other ants. This fact endows the algorithm with high exploration towards a clustering solution. The problem of pheromone memory can also be described as a centroid encoding label problem, in which each cluster is associated with one label. Accordingly, this label change occurs in each iteration following the random selection of the centroid. This issue can be described in the current pheromone memory (Fig. 1), in which two ants produce two clustering solutions  $s$ . The first ant produces  $s_1$  as (1-1-2-2-3-3) and the second ant produces  $s_2$  as (3-3-2-2-1-1). Note that the length of the clustering route is 6, the first instance of solution  $s_1$  is assigned to cluster number 1 and the second instance is assigned to cluster number 1. Both clustering solutions show that each ant produced the same number of groups, in which each group has the same number of members. The difference is that the label of each group differs in each clustering solution. Thus, the pheromone associated with the first object  $o_1$  that belongs to cluster number one (1) of the first solution  $s_1$  differs from solution  $s_2$ , in which the pheromone is associated with cluster number three (3). Accordingly, the pheromone information produced by the first ant becomes useless for the second ant, in which both solutions produce the same result with the same accuracy. This problem will make the algorithm unable to retain the information of the current neighborhood assignment to be used in the next iterations (Fig. 1).

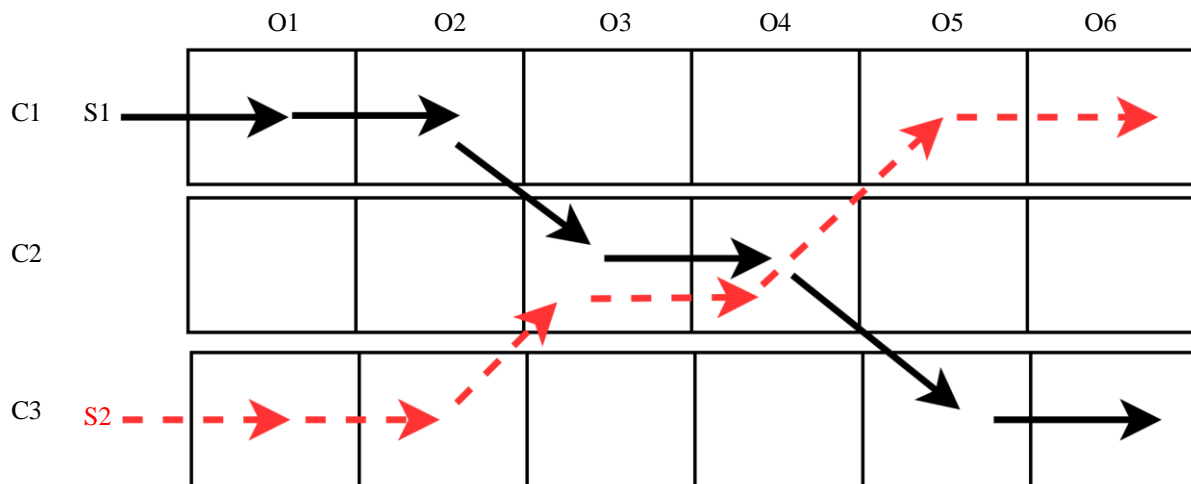


Fig. 1: Construction graphs of ant performing clustering solution based on random centroids

## Research Method

This study attempts to solve the problem of pheromone memory, in which such a memory cannot retain the information of a previous clustering assignment. To illustrate the proposed algorithm, this research considers the number of clusters defined by the user as the parameter for the algorithm. The proposed algorithm relies on only the pheromone which means that no heuristic information is required to construct the clustering solution. To construct a route of attributes, a centroid highly similar to the ant-Miner for the classification problem was employed (Al-Behadili *et al.*, 2019; 2018b) Given the use of an example of a data set, in which the number of attribute  $D$  is three (3), each centroid contains three attributes. The number of clusters  $K$  involves three (3) centroids, in which each single cluster has one single centroid. Thus, the total number of attributes is nine (9) and a centroid is represented by three (3) attributes. The proposed algorithm attempts to optimize the optimal value of each attribute of each centroid. The optimal attribute value is the optimal centroid that can provide the optimal assignment. However, finding the optimal value of each single attribute is complicated because it resembles an NP hard optimization problem. That is, each single attribute may obtain any value between  $[0, 1]$ , in which the minimum value is zero (0) and maximum value is one (1). This characteristic is a continuous problem that is an optimization problem. The proposed algorithm will also optimize the problem to find the best value of each attribute giving the minimum intra-clustering distance. As an example, consider the maximum value of each attribute in the data set as  $[7.9, 4.4, 2.5]$ , where 7.9, 4.4 and 2.5 represent the first, second and third attributes respectively with all attributes having a minimum value of 0.0. Thus, an example of a possible value of the first attribute is between  $[0.0, 7.9]$  and the second attribute is  $[0.0, 4.4]$ .

Accordingly, each attribute should have its own value between the maximum and minimum attribute values. Note that this instance is an example only of a single centroid. Thus, the remaining attributes of the other centroids are produced similarly. The proposed algorithm must identify the optimal value of each attribute according to Equation 1 (Zabihi and Nasiri, 2018):

$$v_{ij} = z_{ij} + \varnothing_{ij} (z_{ik} - z_{ik}) \quad (1)$$

where,  $z_{ij}$  is a real number that represents the minimum value of the current attribute and  $z_{ik}$  is the maximum attribute value.  $\varnothing_{ij}$  is a random number between  $[0, 1]$ . It controls the attribute value between zero (0) and the maximum attribute value. Equation (1) is suitable if the proposed algorithm deals with a co-unions problem, but the said algorithm is a discrete algorithm. Thus, the pheromone memory must be represented by all the possible values that can be carried by  $\varnothing_{ij}$ . In this study, the pheromone memory is the size of  $(P, (K * D))$  where  $P$  is the number of possible values (rows of the pheromone memory) while  $(K * D)$  is the value of the pheromone memory attributes. Figure 2, the value of attributes from 0.1 to 0.5 is discretize to facilitate the understanding of the construction graph of the proposed A-ACOC algorithm. In the graph, each ant travels from one attribute to another, based on the pheromone value. The point that has the maximum pheromone has maximum probability to be selected by the ant. Note that the first point is selected randomly. The ideal route is the route that produces clustering with minimum intra-clustering distance. Figure 2 shows that the route with the bold arrow string contains the route string (0.1-0.2-0.1-0.3-0.3-0.3-0.1-0.5-0.3). If this route string is applied in Equation (1), the result will be three centroids as follows:  $[0.79, 0.88, 0.25]$ ,  $[2.37, 1.32, 0.75]$  and  $[0.79, 2.2, 0.75]$ . Thus, the current ant carries those centroids to construct its clustering solution that produces three (3) groups.

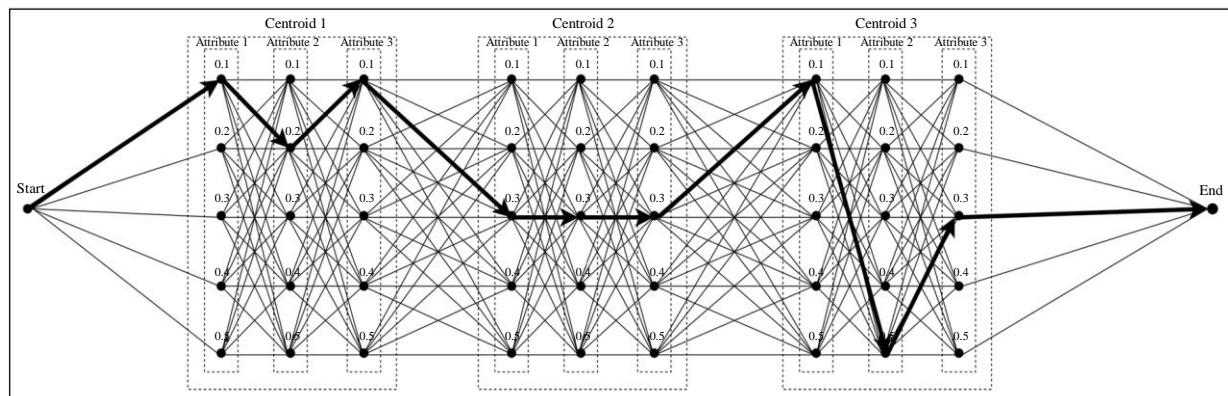


Fig. 2: Construction graph of A-ACOC

**Table 1:** Attribute matrix

Attribute 1	Attribute 2	Attribute 3
0.1	0.1	0.1
0.2	0.2	0.2
0.3	0.3	0.3
0.4	0.4	0.4
0.5	0.5	0.5

Note that in the first run, the value of the pheromone entries are initialized to the same small value and that the value of attributes written in Fig. 2 are stored in another matrix ( $P * D$ ) called the attribute matrix. Each attribute visited in the route produced compensates the corresponding value in the attribute matrix to obtain the final route using Equation (1). Table 1 presents the attribute matrix. Each attribute obtains a different value based on the values of  $P$  that are responsible for the discrete value of each row of the attribute.

**Algorithm Details**

The proposed algorithm assumes that the number of clusters is a predefined parameter known by the user. The centroids of each cluster are extracted from the route constructed by each ant, which means that each ant has its centroids. This outcome differs from those of previous studies (Kao and Cheng, 2006; Santos *et al.*, 2009; Li and Yao, 2016; David and Kosala, 2018), in which the centroids are randomly generated. To construct the clustering solution, the algorithm initializes the pheromone memory of ( $P, (K * D)$ ) and the attribute memory of ( $P * D$ ), where  $P$  is a predefined parameter that represents the number of possible values (rows of the pheromone memory). The value of  $K$  is the number of clusters and  $D$  is the number of attributes. The value of is the value of the pheromone memory columns. The value of the pheromone memory is initialized to a small value and the attributes memory is initialized as  $1/P$ . For example, when  $P$  is equal to the value 10, the first row starts with value 0.1 and gradually increases until the last value becomes 0.9, such as the example used in Table 1. After the initialization of the ants, each ant starts with a random attribute value. The next step is to select the best attribute value on the basis of the exploitation and exploration strategies. Exploitation drives the algorithm to produce a clustering solution using the greedy concept. By contrast, exploration drives the algorithm to produce diversity in the clustering solution to explore the search space. However, both strategies are the engines of the algorithm to produce an optimal clustering solution. Exploitation can be calculated as shown in Equation 2, where  $i$  is the current attribute value  $0 \leq j \leq p$ , is the next attribute value between [0-1] and  $p_{ij}$  is the next attribute in which the value of the attribute is the maximum amount of the pheromone. Note that  $q$  is a random value between [0-1] generated by the algorithm with each selection, while  $q_0$  is a predefined parameter value

between [0-1] and initialized by a user. If  $q$  is greater or equal to  $q_0$ , then the algorithm executes the exploration strategy to select the next attribute value. In this study,  $q_0$  is equal to 0.98. Thus, the probability of the algorithm toward exploitation is higher than that toward the exploration strategy (Equation 3):

$$p_{ij} = \begin{cases} (\tau_{ij})^\alpha & \text{if } q < q_0 \\ S & \text{otherwise} \end{cases} \tag{2}$$

$$p_{ij} = \frac{(\tau_{ij})^\alpha}{\sum_{j=0}^p (\tau_{ij})^\alpha} \tag{3}$$

Equation 3 indicates that the algorithm normalizes the amount of the pheromone to fall in the range [0, 1]. The selection process has a stochastic selection based on the concept of proportionality called Roulette Wheel Selection (RWS). RWS can increase the diversity of the clustering solution by exploring the search space. Note that this approach is crucial in the beginning of the algorithm run, where the algorithm accepted a worse clustering solution in the beginning. This situation endows the algorithm with the ability to avoid being trapped in local optima in the early stages of the algorithm research.

The construction of the centroid route is processed until the tabu list  $tb_r$  of ant  $r$  becomes full. After constructing the centroid route by ant  $r$ , the next step converts this route to the centroid matrix. In this study, each ant has its own centroid matrix that contains ( $K * D$ ), where  $K$  is the number of clusters and  $D$  is the number of attributes (Table 2). Each centroid route is converted using Equation 1 and store in centroid matrix of ant  $r$ . As an example, the route string which is mentioned earlier contains (0.1-0.2-0.1-0.3-0.3-0.3-0.1-0.5-0.3). The produced centroid matrix of ant will be as shown in Table 2. Note that the first three (3) numbers will be stored in the first row of the centroid matrix and the second three (3) numbers will be stored in the second row of the centroid matrix and so on.

The assignment starts when each instance of the dataset is assigned to an appropriate centroid (best cluster) according to the distance between the instance and the centroid matrix. This process is similar to the process of K-means assignment using Equation 4, where  $N$  is the total instances,  $x_i$  is the instance that belongs to  $N$  and  $c_j$  is one of the centroids made available by ant  $r$ . The obtained value of  $u$  is the appropriate cluster where  $1 \leq u \leq K$  (Jain, 2010).

$$u = \min \sum_{i=1}^N \sum_{j=1}^K (x_i - c_j) \tag{4}$$

Step 1: Initialize all parameters  
 Step 2: Iteration 0  
 Step 3: While iteration < Max  
 Step 4: While r < R do  
 Step 5: Construct route centroid by each ant.  
 Step 6: Construct clustering solution by each ant.  
           Select the best ant in a current iteration based on  
           its fitness and perform pheromone update and  
           evaporation process.  
 Step 7: r ++  
 Step 8: Iteration ++  
 Step 9: Print best ant result

**Fig. 3:** Proposed A-ACOC algorithm

**Table 2:** Centroid matrix for each ant of the algorithm

	Attribute 1	Attribute 2	Attribute 3
Centroid 1	0.79	0.88	0.25
Centroid 2	2.37	1.32	0.75
Centroid 3	0.79	2.2	0.75

**Table 3:** Ant clustering solution route

Instances	1	2	3	4	5	6	7	8
Cluster label	2	1	3	2	2	3	2	1

The assignment is processed until all instances are grouped into the  $K$  cluster. The clustering solution produced by each ant  $r$  is similar to a 1D matrix (Table 3). Notably, the first instance of the ant solution is assigned to cluster number 2 and the second instance is assigned to cluster number 1 and so on:

Each ant  $r$  calculates its fitness (i.e., total error within each cluster) to show how close or far each cluster member is to the centroid and how far those clusters are from each other. The total error fitness  $f_r$  is the summation of distance between each instance and ant centroid  $c^r$  and can be calculated as shown in Equation 5. Note that  $d$  is the Euclidean distance between the first number  $x$  and the second number  $y$  which can be computed using Equation 6 (David and Kosala, 2018).

$$f_r = \sum_{i=1}^K \sum_{j=1}^N d(c^r, \text{instance } j), \quad (5)$$

$$d(x, y) = \sqrt{(x - y)^2} \quad (6)$$

After each ant ranks its clustering solution, the algorithm selects the best in the current solution that has the lowest error, which is called the best-iteration-solution. This solution is updated by adding an amount of pheromone. The purpose of this operation is to reward the best ant in its current iteration and so the algorithm guides the ants in the next iterations about that solution with more intensification for greater improvement. The updated pheromone will be implemented on the

centroid's route that produces the best clustering result, as shown in Equation (7) (Shelokar *et al.*, 2004):

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \left(\frac{1.0}{f_r}\right) \quad (7)$$

where,  $\tau_{ij}$  is the amount of pheromone between attribute  $i$  and attribute  $j$  at time  $t$ . The  $\rho$  is the evaporation factor that minimizes the amount of pheromone in the centroid memory in each iteration. This  $\rho$  operation seeks to forget information gathered in previous iterations. Thus, only the best clustering solution frequently updated by the algorithm has a high chance to be alive in the memory for a longer time. This outcome reflects the general process of learning in the ACO, where the feedback of the algorithm plays an important role in the success of algorithms to generate optimal clustering solutions. Note that the algorithm is very sensitive to its parameters such that any change can produce different clustering results, thus the value of the parameters can be tuning as self-adaptive or adaptive to gain a more optimal result. Moreover, the algorithm lacks a local search that can also produce more optimal clustering results. To make the algorithm clearer and more comprehensible to researchers, Figure 3 describes the algorithm.

## Experimental Results

To evaluate the performance of the algorithm, experiments were conducted using 14 standard data sets from the University of California Irvine (UCI) Machine Learning Repository (Bache and Lichman, 2013). Each data set has its own characteristics and the number of clusters varies from one to the other (Table 4). The data sets are from different problems, such as disease that includes Breast Cancer (BC), Breast Tissue (BT), Hepatitis (Hp) and *E. coli* (Ec); ascertaining the survival of patients (Haberman (Hb)); image analysis of the ionosphere (Io); metal signals in mining operations

(Sonar (So)), the iris flower (Ir), fitting contact lenses (Lenses (Le)), mammographic (Mm) analysis of images to distinguish malignant breast disease, measurements of geometrical properties of kernels belonging to three different varieties of wheat (seeds (Se)), Vertebral Column (VC) analysis of orthopedic patients and chemical analysis in wine (Wi). All datasets are of multivariate type, either real or integer. Table 4 lists the distribution of the data sets.

Evaluation against the classical clustering algorithm includes the metaheuristic algorithm and local search algorithm. In the metaheuristic comparison, different clustering algorithms have been used. The algorithms are ACOC, M-ACOC, GA, SA and local search K-means. The number of executions of each algorithm is 10 times, except for the K-means algorithm which is 50 times because this algorithm is easily trapped in the local optima clustering solution. Table 5 presents the parameters of each algorithm. Note that the parameter setting of each algorithm is in a similar domain as reported in Niknam and Amiri (2010).

The clustering result is assessed using the internal evaluation criteria, which is the main objective function of this study and is denoted as the sum of the intra-cluster distances as shown in Equation (5). The quality of the clustering results is considered optimal if the sum of the intra-cluster distances is minimized. Another criterion is used in this research to measure the quality of the clustering results using entropy measurement. Accordingly, the best value is reached if the entropy measurement is minimized (Equation 8).

The entropy values were employed to determine the knowledge of information in the clustering. Equation 8 demonstrates how the entropy for single clustering  $w$  is calculated (Haghir *et al.*, 2008):

$$H(w) = -\sum_{c \in C} P(w_c) \log_2 P(w_c), \quad (8)$$

where,  $c$  is a classification in set  $C$  and  $P(w_c)$  is the probability of a data point being classified as  $c$  in cluster  $w$ . The total entropy of a cluster is as follows:

$$H(\Omega) = \sum_{w \in \Omega} H(w) \frac{N_w}{N}, \quad (9)$$

where,  $H(w)$  is the entropy of a single cluster,  $N_w$  is the number of points in cluster  $w$  and  $N$  is the total number of points. A low total entropy indicates improved clustering results.

Comparisons were performed in two phases. The first phase is a general comparison with the best known (common) clustering algorithms focusing on the internal and external criteria. The internal criterion is the average sum of the intra-cluster distance and the external criterion is the average of the entropy value. The second phase involves the comparison between clustering algorithms that belong to the ACO-based clustering.

The result of the first phase is displayed in Table 6 which shows that A-ACOC obtains the best result in seven (7) data sets (approximately 50%). Joint best performances by GA, SA and KM can be seen on Lo, Se and So datasets. However, among the three (3) algorithms, SA is superior. In the comparison between the algorithm that belongs to the ACO-based cluster, it can be seen from Figure 3 that the A-ACOC obtains the best results for the average intra-cluster distance in 11 data sets (approximately 78%). The second best result is displayed by ACOC in the Lo, Se and So datasets. We can conclude that the proposed algorithm is superior for internal criteria. The proposed algorithm can find the optimal centroids because the learning process of the algorithm depends on previous information which is retained from previous processes.

**Table 4:** Datasets characteristics

Dataset Name	Attributes	Classes	Instances
Breast cancer	9	2	699
Breast tissue	9	6	106
E. coli	7	6	336
Haberman	3	2	306
Hayes	5	6	132
Hepatitis	19	2	155
Ionosphere	34	2	351
Iris	4	3	150
Lenses	4	3	24
Mammographic	5	2	961
Seeds	7	3	210
Sonar	60	2	208
Vertebral column	6	3	310
Wine	13	3	178

**Table 5:** Values of the parameters for each algorithm

Algorithm	GA		SA		ACOC		A-ACOC/ M-ACOC	
	Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
	Population	50	Probability threshold	0.98	Ants	50	Ants	50
	Crossover	0.8	Initial temperature	5	Probability threshold	0.98	Probability threshold	0.001
	Mutation rate	0.001	Temperature multiplier	0.98	Local search rate	0.01	Local search rate	0.01
	Iterations	10	Final temperature	0.001	Evaporation rate	0.01	Evaporation rate	0.01
		Iterations	1000	Iterations	1000	Iterations	1000	

The second comparison focuses on the external criteria, i.e., minimum value of entropy. Results are shown in Table 7 for all common clustering algorithms which show the absence of a dominant algorithm. Figure 4 displays the results of ACO-based algorithms for external criteria. Joint best results were obtained by the three (3) algorithms for the Lo, Hp and Hb datasets. However, among the three (3) algorithms, A-ACOC is more superior because it obtained another five (5) best results.

algorithms were design based on the internal criteria and, thus, no dominant algorithm can be seen in this experiment.

Figure 5 displays the behavior of A-ACOC in performing the clustering task for three (3) experimental settings on Wi dataset. The number of ants used in the first, second and third runs are 250, 500 and 1000. The algorithm starts with high exploration followed by the long exploration period which shows how the search processes are performed.

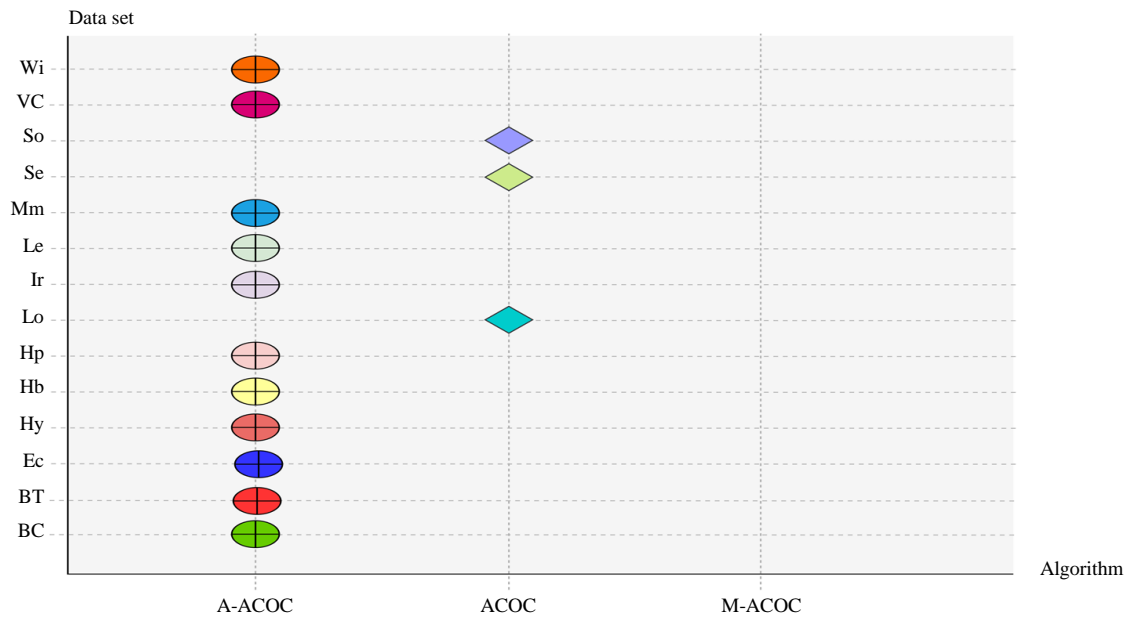


Fig. 3: Average intra-cluster distance for ACO-based clustering algorithms

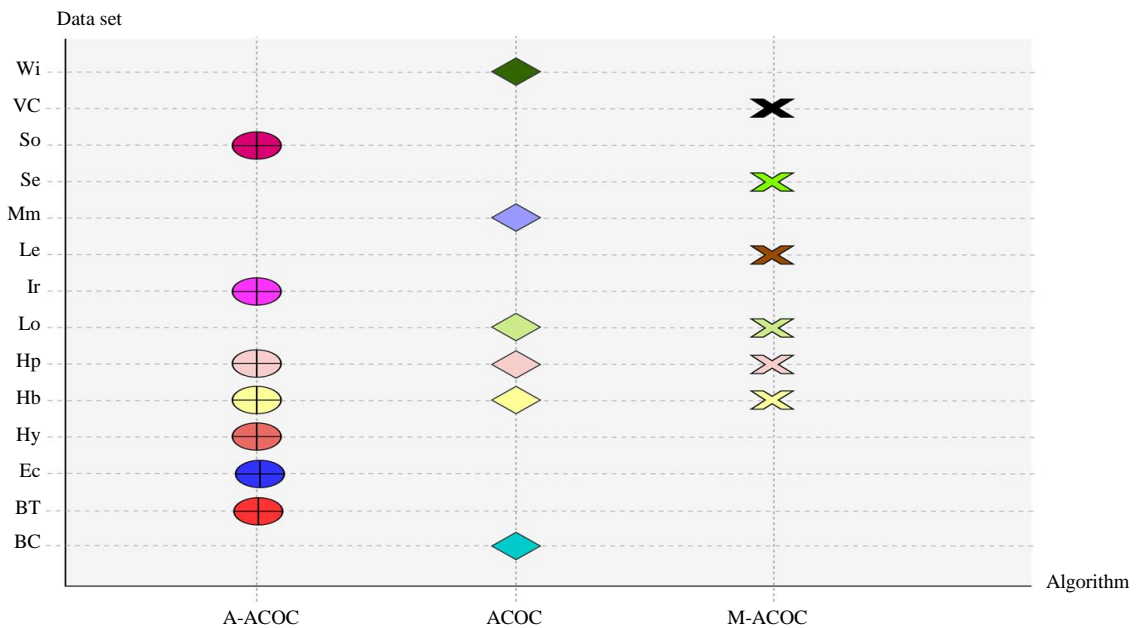
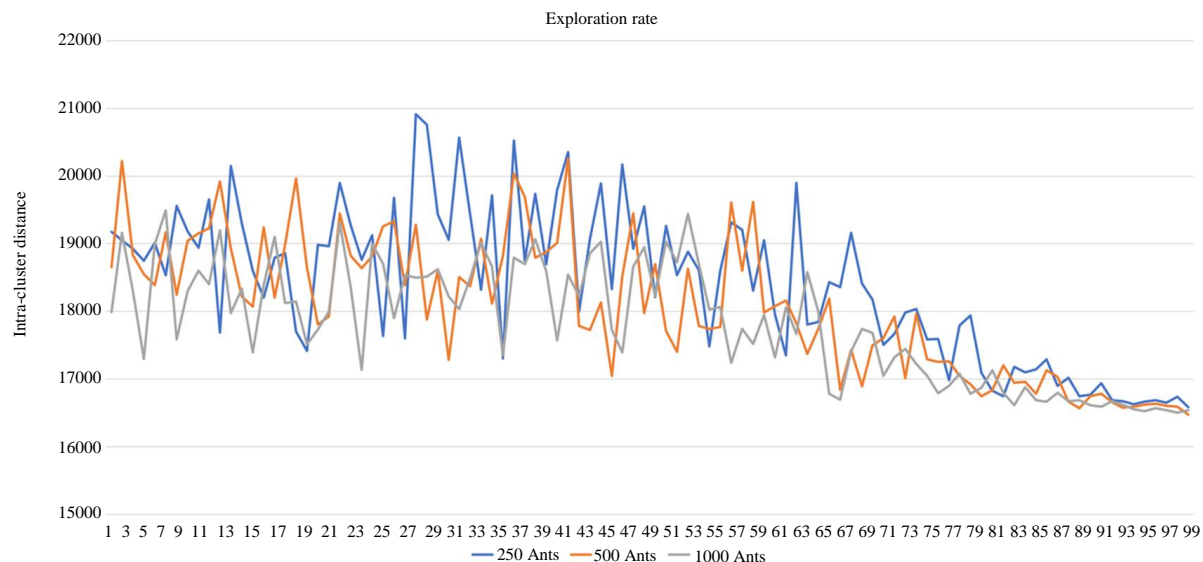


Fig. 4: Average entropy result for ACO-based clustering algorithms





**Fig. 5:** Behavior pattern of A-ACOC algorithm

**Table 6:** Average intra-cluster distance for common clustering algorithms

Dataset	Algorithms					
	A-ACOC	ACOC	M-ACOC	GA	SA	KM
Breast cancer (BC)	3044.801	4196.055	3213.651	3055.420	3061.344	3055.64
Breast tissue (BT)	7161.515	7222.337	8081.864	7130.35	7054.088	7082.394
Ecoli (Ec)	71.134	71.550	74.136	69.801	70.249	67.953
Haberman (Hb)	2568.964	2625.918	2601.155	2625.585	2625.107	2625.290
Hayes (Hy)	142.832	144.854	152.362	145.778	142.947	143.940
Hepatitis (Hp)	8717.622	8831.952	8948.949	8853.492	8853.492	8854.854
Ionosphere (Lo)	809.870	796.336	803.280	796.044	796.044	796.055
Iris (Ir)	97.028	97.167	99.193	97.332	97.222	97.229
Lenses (Le)	20.787	21.798	24.727	20.884	20.784	20.876
Mammographic (Mm)	6969.332	7779.805	7023.550	7034.043	7216.227	7033.208
Seeds (Se)	316.622	315.188	315.596	313.216	313.216	313.216
Sonar (So)	249.054	234.753	258.363	234.740	234.740	234.742
Vertebral column (VC)	7958.531	7969.347	8091.551	7856.606	7846.458	7860.869
Wine (Wi)	16309.284	16525.214	16427.749	16530.537	16530.537	16540.328

**Table 7:** Average entropy result for common clustering algorithms

Dataset	Algorithm					
	A-ACOC	ACOC	M-ACOC	GA	SA	KM
Breast cancer (BC)	0.230	0.224	0.280	0.214	0.221	0.249
Breast tissue (BT)	2.039	2.052	2.104	2.049	1.981	2.060
Ecoli (Ec)	0.719	0.805	0.809	0.812	0.837	0.747
Haberman (Hb)	0.833	0.833	0.833	0.832	0.832	0.830
Hayes (Hy)	1.284	1.295	1.371	1.259	1.293	1.284
Hepatitis (Hp)	0.733	0.733	0.733	0.734	0.734	0.732
Ionosphere (Lo)	0.828	0.816	0.816	0.814	0.814	0.811
Iris (Ir)	0.380	0.396	0.390	0.380	0.379	0.466
Lenses (Le)	1.286	1.202	1.035	1.191	1.145	1.163
Mammographic (Mm)	0.901	0.893	0.898	0.894	0.906	0.901
Seeds (Se)	0.484	0.526	0.473	0.485	0.485	0.486
Sonar (So)	0.987	0.990	0.996	0.990	0.990	0.988
Vertebral column (VC)	0.873	0.917	0.839	0.831	0.864	0.882
Wine (Wi)	0.893	0.889	0.891	0.909	0.909	0.911

## Discussion

This section explores the clustering results and the proposed memory strategy. The proposed algorithm outperforms all best-known clustering algorithms in the minimum average of intra-cluster distance but is at par with other clustering algorithms in the minimum value of entropy measurement. Entropy measurement is external clustering calculated on the basis of information that reflects real data classification. Conversely, the objective function of all the clustering algorithms in this study is the internal clustering calculated based on internal information according to the minimum intra-clustering variance.

Finding the optimal centroid has been solved by the proposed algorithm which uses the adaptive instance route instead of the random strategy that has been used by other algorithms. This was possible because the process of finding the optimal results is based on the process of finding best centroids using only the pheromone.

The learning process in the proposed algorithm is better than in other algorithms because the proposed algorithm was able to control the pheromone update better than the algorithms that belong to the ACO-based clustering. The reason for this is because the learning process was successfully applied when each clustering solution was rewarded according to fitness. The ACO-based clustering algorithm fails to identify the centroids in an explicit way.

The proposed algorithm identifies the centroids based on the centroid's route and intensifies the search in the neighborhood of the centroids by using both exploration and exploitation strategies. Thus, further exploration of the neighborhood helps to identify more clustering results with lower intra-distances.

## Conclusion

This study has addressed the problem of finding the optimal initial cluster center (centroids) to avoid being trapped in the local minima problem. The improvement is achieved using a new and improved pheromone memory as the centroid's memory. The algorithm identifies optimal centroids by finding the optimal centroid's route that minimizes the clustering problem. The proposed algorithm finds the optimal centroids on the basis of the amount of pheromone without any involvement of heuristic information. This approach served to control the decentralization aspect of the proposed algorithm.

The performance of the proposed A-ACOC was tested using 14 standard data sets taken from the UCI Machine Learning Repository. The proposed algorithm outperformed other common clustering algorithms based on minimum intra-clustering distance. Future research will focus on evaluating the proposed algorithm on other datasets using other evaluation criteria.

## Acknowledgement

The authors thank the Ministry of Higher Education Malaysia for funding this study under the Transdisciplinary Research Grant Scheme (S/O code 14163).

## Author's Contribution

All authors equally contributed to this research, including in the analysis of the results, organization of writing and preparation of the manuscript.

## Ethics

This research is original and contains unpublished materials. The corresponding author confirms that all other authors have read and approved the manuscript and no ethical issues are involved.

## References

- Abuhamdah, A.F., 2018. Adaptive elitist-ant system for medical clustering problem. *J. King Saud University - Computer Information Sciences*. DOI: 10.1016/j.jksuci.2018.08.007
- Al-Behadili, H.N.K., 2018. Intelligent hypothermia care system using ant colony optimization for rules prediction. *J. University Babylon*, 26: 47-56. DOI: 10.29196/jub.v26i2.473
- Al-Behadili, H.N.K., K.R. Ku-Mahamud and R. Sagban, 2018b. Rule pruning techniques in the ant-miner classification algorithm and its variants: A review. *Proceedings of the IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE)*, IEEE Xplore press, pp: 47-56. DOI: 10.1109/ISCAIE.2018.8405448
- Al-behadili, H.N.K., K.R. Ku-Mahamud and R. Sagban, 2019. Annealing strategy for an enhance rule pruning technique in ACO-based rule classification. *Indonesian J. Electrical Engineering Computer Sci.*, 16: 1499-1507. DOI: 10.11591/ijeecs.v16.i3.pp1499-1507
- Al-Behadili, H.N.K., K.R. Ku-Mahamud and R. Sagban, 2018a. Ant colony optimization algorithm for rule-based classification: Issues and potential solutions. *J. Theoretical Applied Information Technol.*, 96: 7139-7150.
- Bache, K. and M. Lichman, 2013. *UCI machine learning repository*. University of California Irvine School of Information and Computer Sciences.
- Binu, D., 2015. Expert systems with applications cluster analysis using optimization algorithms with newly designed objective functions. *Expert Systems Applications*.

- David and R.R. Kosala, 2018. Clustering algorithm comparison of search results documents. Proceedings of the 6th International Conference on Cyber and IT Service Management (CITSM), pp: 1-6. DOI: 10.1109/CITSM.2018.8674246
- Haghir, M., H. Abolhassani and M. Haghir, 2008. Improving density-based methods for hierarchical clustering of web pages. *Data Knowledge Eng.*, 67: 30-50.
- Jabbar, A.M., 2018. Controlling the balance of exploration and exploitation in ACO algorithm. *J. University Babylon*, 26: 1-9.
- Jabbar, A.M., K.R. Ku-Mahamud and R. Sagban, 2018. Ant-based sorting and ACO-based clustering approaches: A review. Proceedings of the IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE), pp: 217-223.
- Jain, A.K., 2010. Data clustering: 50 years beyond K-means. *Pattern Recognition Lett.*, 31: 651-666.
- Kao, Y. and K. Cheng, 2006. An ACO-based clustering algorithm. ANTS: International Workshop on Ant Colony Optimization and Swarm Intelligence, pp: 340-347.
- Kumar, Y. and G. Sahoo, 2014. A chaotic charged system search approach for data clustering. *Informatica (Slovenia)*, 38: 249-261. DOI: 10.1007/s13748-014-0049-2
- Li, M. and M. Yao, 2016. An optimized ant system for clustering with elitist ant and local search. Zhejiang University, Hang Zhou, China.
- Murtagh, F., 1983. A survey of recent advances in hierarchical clustering algorithms. *Computer J.*, 26: 354-359.
- Nagpal, A., A. Jatain and D. Gaur, 2013. Review based on data clustering algorithms. Proceedings of the IEEE Conference on Information Communication Technologies (ICT), pp: 298-303.
- Niknam, T. and B. Amiri, 2010. An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis. *Applied Soft Computing J.*, 10: 183-197.
- Popat, S.K. and M. Emmanuel, 2014. Review and comparative study of clustering techniques. *Int. J. Computer Sci. Information Technologies*, 5: 805-812.
- Santos, D.S., D. de Oliveira and A.L. Bazzan, 2009. Data mining and multi-agent integration. *Data Mining and Multi-Agent Integration*, pp: 239-249.
- Shabanzadeh, P. and R. Yusof, 2015. An efficient optimization method for solving unsupervised data classification problems. *Computational Mathematical Methods Med.* DOI: 10.1155/2015/802754
- Shelokar, P.S., V.K. Jayaraman and B.D. Kulkarni, 2004. An ant colony approach for clustering. *Analytica Chimica Acta*, 2: 187-195.
- Singh, H., Y. Kumar and S. Kumar, 2019. A new meta-heuristic algorithm based on chemical reactions for partitional clustering problems. *Evolutionary Intelligence*, DOI: 10.1007/s12065-019-00221-w
- Stutzle, T. and S. Linke, 2000. Experiments with variants of ant algorithms. *Mathware Soft Computing*, 7: 1-14.
- Ünlü, R. and P. Xanthopoulos, 2019. Estimating the number of clusters in a dataset via consensus clustering. *Expert Systems Applications*, 125: 33-39. DOI: 10.1016/j.eswa.2019.01.074
- Wahid, J. and H.F.A. Al-Mazini, 2018. Classification of cervical cancer using ant-miner for medical expertise knowledge management. Proceedings of the Knowledge Management International Conference (KMICe).
- Zabihi, F. and B. Nasiri, 2018. A novel history-driven artificial bee colony algorithm for data clustering. *Applied Soft Computing J.*, 71: 226-241. DOI: 10.1016/j.asoc.2018.06.013
- Zhang, C. and S. Xia, 2009. K-means clustering algorithm with improved initial center. Proceedings of the 2nd International Workshop on Knowledge Discovery and Data Mining, (WKKD, 09), pp: 790-792.