

An improved ACS algorithm for data clustering

Ayad Mohammed Jabbar¹, Ku Ruhana Ku-Mahamud², Rafid Sagban³

¹College of Arts and Sciences, Shatt Al-Arab University, Iraq

²Data Science Research Lab, School of Computing, Universiti Utara Malaysia, Malaysia

³Computer Science Department, University of Babylon, Iraq

Article Info

Article history:

Received Jul 7, 2019

Revised Sep 10, 2019

Accepted Sep 26, 2019

Keywords:

Ant colony optimisation

Data clustering

Data mining

Optimisation based-clustering

Swarm intelligence

ABSTRACT

Data clustering is a data mining technique that discovers hidden patterns by creating groups (clusters) of objects. Each object in every cluster exhibits sufficient similarity to its neighbourhood, whereas objects with insufficient similarity are found in other clusters. Data clustering techniques minimise intra-cluster similarity in each cluster and maximise inter-cluster dissimilarity amongst different clusters. Ant colony optimisation for clustering (ACOC) is a swarm algorithm inspired by the foraging behaviour of ants. This algorithm minimises deterministic imperfections in which clustering is considered an optimisation problem. However, ACOC suffers from high diversification in which the algorithm cannot search for best solutions in the local neighbourhood. To improve the ACOC, this study proposes a modified ACOC, called M-ACOC, which has a modification rate parameter that controls the convergence of the algorithm. Comparison of the performance of several common clustering algorithms using real-world datasets shows that the accuracy results of the proposed algorithm surpasses other algorithms.

*Copyright © 2020 Institute of Advanced Engineering and Science.
All rights reserved.*

Corresponding Author:

Ayad Mohammed Jabbar,

College of Arts and Sciences,

Shatt Al-Arab University, Basra, Iraq.

Email: ayadmohammed@sa-uc.edu.iq

1. INTRODUCTION

Clustering applies cluster analysis to organise data as clusters. Similarities are measured on the basis of the extracted features amongst data. This indirect data mining approach performs clustering without using predefined classes (unlabelled data) to determine the relationship amongst data, whereas a direct approach (classification) requires the use of predefined classes (pre-labelled data) [1, 2]. The clustering approach groups data into different clusters that contain similar objects on the basis of an appropriate fitness measure that determines the relationship amongst data [3-5]. The clustering process has four major steps [6, 7]. Firstly, the features of a set of candidates are extracted from the data. Secondly, an appropriate proximity measure is selected on the basis of the data required for clustering [8, 9]. Thirdly, the cluster results are validated on the basis of testing criteria. Lastly, the clustering results are interpreted to verify the reliability of useful information. Data clustering can be classified into deterministic and stochastic approaches. The major drawback of the deterministic approach is the difficulty in determining the optimal number of clusters that are required as a predefined value and getting stuck in local optima [10, 11]. Given the shortcoming of this approach, researchers have focused on using the metaheuristic approach, which is inspired by insects and their natural behaviour. The metaheuristic approach uses a completely different clustering method wherein the clustering problem is formulated as an optimisation problem [12-14]. This approach minimises or maximises an objective function to find the maximum similarity amongst data [15-17]. Although the metaheuristic approach does not guarantee an optimal clustering solution, it can find good solutions within a

relatively short time in practice. This approach performs clustering on the basis of an individual or a population [18-20].

Ant colony optimisation (ACO) is a metaheuristic algorithm based on the foraging behaviour of ants. It is used in different data mining application domains such as classification and clustering [21, 22]. Kao and Cheng [19] introduced dynamic cluster centres, namely ACOC is based on the assumption that the number of clusters is fixed. The ACOC uses the concept of an ant colony system (ACS) for clustering problems. The major shortcoming of this algorithm is the slow convergence towards the best clustering solution [23-25]. This algorithm exhibits high diversification and, thus, it performs poorly in searching the local neighbourhood for best solutions, where the incapable algorithm moves the search to best region in the search space [26].

The current study proposes a new modification parameter, called the modification rate (MR) and convergence rate (CR), where the MR controls the algorithm convergence based on the value of the CR . The MR is used to force the algorithm to search the local neighbourhood, thereby enabling it to intensify the search for best solutions in the neighbourhood before moving to another location. The MR moves the algorithm search to a new location when the algorithm is stuck in local optima. The MR value, which is statically initialised, is the key to this proposed method. This value determines the CR of the algorithm when the similarity of clustering solutions is high during an algorithm run. Thus, numerous optimal clustering results can be produced.

2. RESEARCH METHOD

The proposed modification is on the intensification process of the ants in an ACS algorithm. To improve the algorithm's performance, this study increases ant intensification of the clustering solution to find more optimal clustering results in the local neighbourhood of best solutions. The ACOC algorithm exhibits a shortcoming in intensification strategy. The algorithm demonstrates high diversification that causes it to perform poorly in searching the local neighbourhood of best solutions. Thus, the ACOC cannot move the search to the best regions in the search space. This study proposes two modifications to overcome the setback of ACOC. The first proposed modification involves a new parameter, the MR and CR , which controls convergence of the algorithm. The proposed M-ACOC has a new archive (known as P) which stores new populations of solutions, the archive P is used to memorise a certain number of solutions, n , in the history of iterations during an algorithm run. This short-term memory memorises the modern history of the search. Thus, the P archive is updated iteratively with the latest clustering solutions by adopting a first in, first out queue behaviour [27]. The best clustering solution in each iteration, which is called iteration-best solution, enters the P archive. If the archive is full, then the new solution replaces the oldest solution in P . The CR represents the amount of similarity/diversity of the solutions located in the P archive. If standard division of each single solution exceeds the prescribed CR , then the value of MR is updated. The algorithm moves the search into another region when the MR value exceeds the limit. This improves the algorithm's performance by exploring new solutions. Notably, the CR parameter, the limit of MR and the size of the P archive are initialised off-line, and the algorithm centroids are updated in accordance with the MR parameter, which is based on the convergence condition of the algorithm.

3. THE PROPOSED ALGORITHM

The proposed algorithm has four major components: the P archive, MR , CR and mating process. The algorithm generates a clustering solution at the beginning of the algorithm. The best-iteration-solution enters the P archive during the algorithm run.

The size of the P archive is initialised off-line. This archive affects both the convergence of the algorithm and CR during the algorithm run. When the P archive is full, the algorithm checks the convergence by calculating the similarity/diversity of the solutions located in the archive. If the diversity of solutions are low, the value of standard division of each single solution will be bigger than the CR and vice versa. The MR parameter value is updated based on similarity/diversity of the solutions. If the MR value exceeds the permissible limit, then the mating process will be performed to explore new clustering solutions in the neighbourhood of best solutions. The new clustering solution will be generated randomly and will be mated with the best-so-far solution that has been obtained during the algorithm run. Thus, this will make the algorithm look for a new clustering solution around the best region in the search space. The algorithm centroids are controlled by the MR . Each time the algorithm begins to converge on the same clustering solution, the mating process is performed to change the behaviour of the algorithm from exploitation to exploration. This will enable the converging process of the algorithm to be flexible during the algorithm run.

This is very important at the beginning of the search because it helps the algorithm to search multiple regions before moving to the next region. The process flow of the M-ACOC is illustrated in Figure 1.

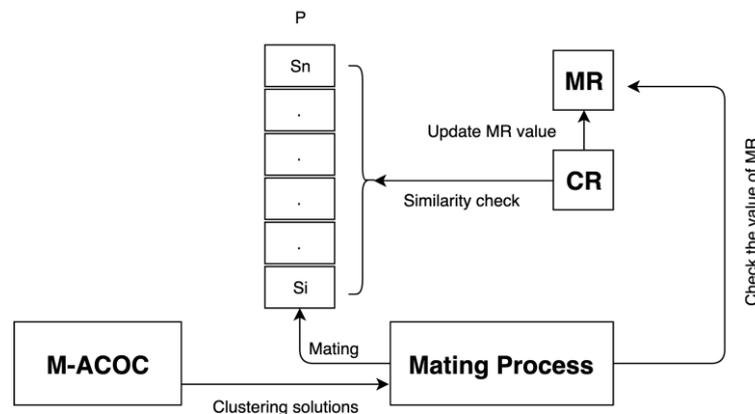


Figure 1. Process flow of M-ACOC

The algorithm begins with random centroids c_{ij}^r , where each ant r has its own centroid matrix and each attribute j of this ant centroid is unique. Meanwhile, i represents a k cluster. Each ant constructs its solution by assigning each object o to the nearest centroid c_{ij}^r . k denotes the number of clusters known and initialised as an off-line parameter by the user. In each step of the assignment, an ant selects one of the k clusters for each object o based on one of two strategies: diversification or intensification. In intensification, an ant greedily selects an appropriate k which is called a pseudo proportional rule [28]. The appropriate k cluster is based on a predefined parameter q_0 , which identifies the intensification magnitude. The minimum value is 0 while the maximum value is 1; the maximum value indicates that the algorithm fully performs intensification as defined in (1).

$$0 \leq q, q_0 \leq 1 \tag{1}$$

The algorithm dynamically generates a random number called q as defined in (1). The intensification strategy is used when $q < q_0$ as shown in (2). The appropriate k is selected based on the amount of pheromone and distance.

$$k = \begin{cases} \arg \max_{c_{ij}^r \in N_o} \{ [\tau(o, c_{ij}^r)^\alpha] [\eta(o, c_{ij}^r)^\beta] \} & \text{if } q < q_0, \\ S & \text{otherwise} \end{cases} \tag{2}$$

where $\tau(o, c_{ij}^r)^\alpha$ is the pheromone level between object o and ant centroid c_{ij}^r . α is a weight parameter measure that specifies the relative weight of the pheromone amount, and $\alpha > 0$. $\eta(o, c_{ij}^r)^\beta$ is the heuristic value of object o and the ant centroid of cluster c_{ij}^r . It is calculated as $\frac{1.0}{d(o, c_{ij}^r)}$, where d is the distance between object o and ant centroid c_{ij}^r . Parameter β specifies the relative weight of the heuristic, and $\beta > 0$. The distance measurement used in this study is the Euclidean distance, which can be calculated as shown in (3) [29].

$$d(o, c_{ij}^r) = \sqrt{(\text{object } o - c_{ij}^r)^2} \tag{3}$$

In exploration, S is selected using (4).

$$p(k) = \frac{[\tau(o, c_{ij}^r)^\alpha][\eta(o, c_{ij}^r)^\beta]}{\sum_{i=1}^k [\tau(o, c_{ij}^r)^\alpha][\eta(o, c_{ij}^r)^\beta]} \tag{4}$$

The assignment is processed until the tabu list tb_r of ant r is full. Each ant r constructs its own clustering solution C_r , which is a 1D matrix, as shown in Figure 2. The fitness f_r of the clustering can be calculated as the minimum error between each object o and the ant centroid of cluster c_{ij}^r . Fitness f_r can be calculated as shown in (6). Then, each ant r updates its weight matrix w_{ij}^r and it's centroid c_{ij}^r that will be used in the next iteration. w_{ij}^r and c_{ij}^r can be calculated as shown in (5) and (7), respectively. w_{ij}^r is a 2D matrix, where $0 < i \leq k$ and $j = N$. N is the number of objects o in the dataset.

$$w_{ij}^r = \begin{cases} 1 & \text{if object } j \text{ belong to cluster } i \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

For example, if ant r constructs a clustering solution as shown in Figure 2, where each object o of the first row is represented by one corresponding k cluster in the second row, then each object belongs to a unique single cluster. Notably, the first object o of the ant r solution is assigned to cluster number 2, and the second object o is assigned to cluster number 1 and so on. A shown in (5), w_{ij}^r represents each object o as a binary value, either 0 or 1, as shown in Table 1, where $k = 3$.

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

Table 1. Weight Matrix

N	1	2	3	4	5	6	7	8
k	1	0	1	0	0	0	0	1
	2	1	0	0		0	1	0
	3	0	0	1	0	0	1	0

Figure 2. Ant clustering solution route

Fitness f_r refers to the minimum error between each object and the k cluster ant centroid c_{ij}^r , which can be calculated as shown in (6). Each ant has its own centroid c_{ij}^r at the beginning of the algorithm run.

$$f_r = \sum_{i=1}^k \sum_{j=1}^N w_{ij}^r * d(c_{ij}^r, \text{object } j) \tag{6}$$

The new centroid of each cluster centre is updated for each ant r based on (7).

$$c_{ij}^r = \frac{\sum_{j=1}^N \text{object } j * w_{ij}^r}{\sum_{j=1}^N w_{ij}^r} \tag{7}$$

Where,

$$i = 1, \dots, k$$

After ranking each ant r based on its fitness f_r , the algorithm performs local search on the best ant at the current iteration, which has the lowest error. Local search is performed using the random selection strategy when the selection of neighbours is based on random values generated for each object of the best-iteration-solution with a predefined local search parameter, called p_{ls} , within the range $0 \leq p_{ls} \leq 1$. In this study, p_{ls} is initialised with a value of 0.01. Thus, only the value of a random number that corresponds to the objects less than p_{ls} will be assigned a different cluster number. If the solution is improved, then it will be accepted and the centroids will update for the current ant; otherwise, it will be rejected. The next process is updating the ant trail solution with an amount of pheromone based on the value of fitness f_r and then performing evaporation on all the entries of the pheromone matrix in accordance with (8).

$$\tau_{ij}(t + 1) = (1 - \rho)\tau_{ij}(t) + \left(\frac{1}{f_r}\right), \tag{8}$$

where $i = 1, \dots, k$; $j = 1, \dots, N$ and f_r is the current fitness of the best ant r in this iteration. ρ is the evaporation rate, and a high ρ value suggests that the information gathered in the previous iterations is quickly forgotten.

In the M-ACOC algorithm, the control parameter MR is added to produce a new clustering solution C_r for ants during the search process in accordance with (9).

$$c_{ij}^r = \begin{cases} \text{Mating} (c_{ij}^r_{\text{best-so-far solution}}, c_{ij}^r_{\text{random}}) & \text{if } MR < p_{ls} \\ \text{ibs} & \text{otherwise} \end{cases}, \tag{9}$$

where c_{ij}^r is the best-so-far solution centroid obtained by the algorithm during the run, and c_{ij}^r is a random centroid generated by the algorithm; otherwise, the algorithm will continue with its iteration-best-solution ibs . The mating process refers to the average of the randomly generated centroids and best-so-far solution centroid. Thus, the algorithm intensifies the search in the location of the best-so-far solutions. Notably, this process is performed by all the ants when the P archive contains solutions with low diversity. Thus, the next iteration for all the ants begins with a new solution obtained from (9). In the current study, the limit of the MR value is equal to 4. This work calculates the standard deviation (SD) of all existing solutions in the P archive to measure their similarity as shown in (10). Each single solution S_i is considered a doubled solution if its SD is less than or equal to the CR . The latter is equal to 0.25 in this study and can be calculated in accordance with (11).

$$SD = \text{Average } P(S_i) - S_i, \quad (10)$$

where
 $i = 0, 1, 2, \dots, P_{Size}$.

$$\text{If } (SD \leq CR) \rightarrow MR = +1, \quad (11)$$

where
 $SD \geq 0$ and $1 \leq MR \leq P_{Size}$.

The CR and MR are the most important parameters for algorithm convergence because the CR identifies the amount of similarity. Thus, if the value of the algorithm is small, then it performs a quick mating process because the MR will be updated immediately. By contrast, when the CR has a large value, the mating process will be slow. Thus, the algorithm converges slowly. The MR works in a similar way to the CR because it identifies the similarity level in the P archive. The M-ACOC algorithm is illustrated as shown in Figure 3.

```

Initialise M-ACOC parameters
while iteration < Max Iteration do
  while r < R do
    Send r ants each with own random centroids
    Construct solution  $s_i$  using pheromone amount and heuristic based on ant's centroid
    Calculate fitness of each ant r using its centroid
    Selects best r ant out of R as best ant in this current iteration and performs local search on this solution
    Update pheromone memory using the best r ant in this current iteration
    Performed evaporation on the pheromone matrix
    Update ants centroid of each ant r
  r = r + 1, Finish ants loop when r = R
  Performs exploration strategy when MR value exceeds the limit
iteration = iteration + 1

```

Figure 3. Proposed M-ACOC algorithm

4. RESULTS AND DISCUSSION

The performance of the M-ACOC algorithm to solve data clustering problems has been evaluated based on six different standard datasets that are most commonly used in the clustering evaluation domain. Performance analysis of the proposed M-ACOC was done on six datasets of numerical type. These datasets are iris, glass, liver disease (LD), thyroid, contraceptive method choice (CMC) and wine; they were taken from the University of California (UCI) Irvine Machine Learning Repository [30]. The characteristics of the datasets are provided in Table 2. The datasets include different features of problem space such as sample dimension, feature diversity, size (small, medium, large), clusters layout (dispersion), the range of changes in various dimensions of the dataset and various number of classes.

Table 2. Characteristics of Datasets

Dataset	Classes	Features	Total objects	Object in each class
Iris	3	4	150	(50, 50, 50)
Glass	6	9	214	(70, 17, 76, 13, 9, 29)
LD	2	6	345	(145, 200)
Thyroid	3	3	215	(150, 30, 35)
CMC	3	9	1473	(629, 334, 510)
Wine	3	13	178	(59, 71, 48)

The parameter settings for the M-ACOC algorithm with the other four algorithms are provided in Table 3. It is noted that those parameters are best known for the setting of those algorithms. The population in GA is important where the initialisation of GA individuals affects the algorithm’s performance. Crossover and mutation are related to the diversity of clustering solution and jump of local minima. Swarm size affects the performance of PSO. If the number of particles is too few the algorithm prompts through the local optimal solutions; meanwhile, if the number of particles is too many the algorithm is slowed down. The value of the maximum velocity ω_{min} and ω_{max} affects algorithm performance in terms of exploration and exploitation, if the value is too high, the algorithm produces clustering solutions without enough exploration of search space, whereas if the value of velocity is too small, the algorithm is suffering to converge at the right time. In ant colony optimisation for clustering, the probability threshold is responsible for the diversity of the clustering solution while local search rate is responsible for the number of updated performances on best clustering solution. The evaporation rate of pheromone can be seen as an exploration mechanism that delays faster convergence of all ants towards a suboptimal path. Table 3 provides the parameter setting of each algorithm.

Table 3. Values of the Parameters for Each of the Five Algorithms

GA		PSO		Shelokar-ACO		ACOC/M-ACOC	
Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
Population	50	# Swarm	10 x K x d	# ants	50	# ants	50
Crossover	0.8	$c_1 = c_2$	2	Probability threshold	0.98	Probability threshold	0.001
Mutation rate	0.001	$\omega_{min} = \omega_{max}$	1	Local search rate	0.01	Local search rate	0.01
# iterations	1000	# iterations	500	Evaporation rate	0.01	Evaporation rate	0.01
				# iterations	1000	# iterations	1000

Two evaluation criteria are used in this study. These criteria include internal criteria which is the sum of intra-cluster distances, i.e., the fitness f_r that has been used earlier in (6); and external criteria which is the F-measure. The sum of the intra-cluster distances measure the summation of error within each cluster. The F-measure is applied to identify the quality of clustering results and the best value if maximised. It requires two additional measure criteria to calculate its results as shown in (12). These measures are called precision and recall, which can be calculated as shown in (13) and (14), respectively [31].

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{12}$$

$$Precision = \frac{TP}{TP + FP}, \tag{13}$$

$$Recall = \frac{TP}{TP + FN}, \tag{14}$$

Where TP is the true positive, FP is the false positive and FN is the false negative.

The sum of intra-cluster distances can be defined as the distances between objects placed in the same cluster that corresponds to the cluster centre c_{ij}^r . The results are measured in terms of the best result (minimum sum of intra-cluster distances) and maximum value of F-measure. Clustering quality is directly related to the minimum sum of distances, whereas clustering accuracy is determined using the F-measure, as shown in Figures 4 and 5, respectively. In both figures, the M-ACOC algorithm is compared with the k -means, genetic algorithm (GA), particle swarm optimisation (PSO), Shelokar-ACO and ACOC in terms of the parameters listed in Table 3, where each algorithm is performed 10 times and the maximum number of

iterations is equal to 1000 [32]. These popular swarm algorithms are the most commonly used in the solutions to a specific problem of clustering evaluation.

Figure 4 depicts the results for sum of intra-cluster distances for five swarm clustering algorithms which have been applied on six datasets. Lower intra-cluster distance indicates minimum error between the data within the cluster. The proposed M-ACOC algorithm outperforms the other clustering algorithms on five datasets (about 83%), namely, iris, wine, CMC, thyroid and glass. PSO obtains the best results for the LD dataset. This is because the M-ACOC algorithm is based on the centroid value.

The F-measure metric is used to show the accuracy of the clustering, in which case, the higher the value, the more accurate is the clustering process. In Figure 5, the M-ACOC outperforms other algorithms on three datasets (iris, LD and glass) while the ACOC obtains the highest F-measure on three datasets (wine, CMC and thyroid). The performance of the M-ACOC and ACOC are comparable for the F-measure metric.

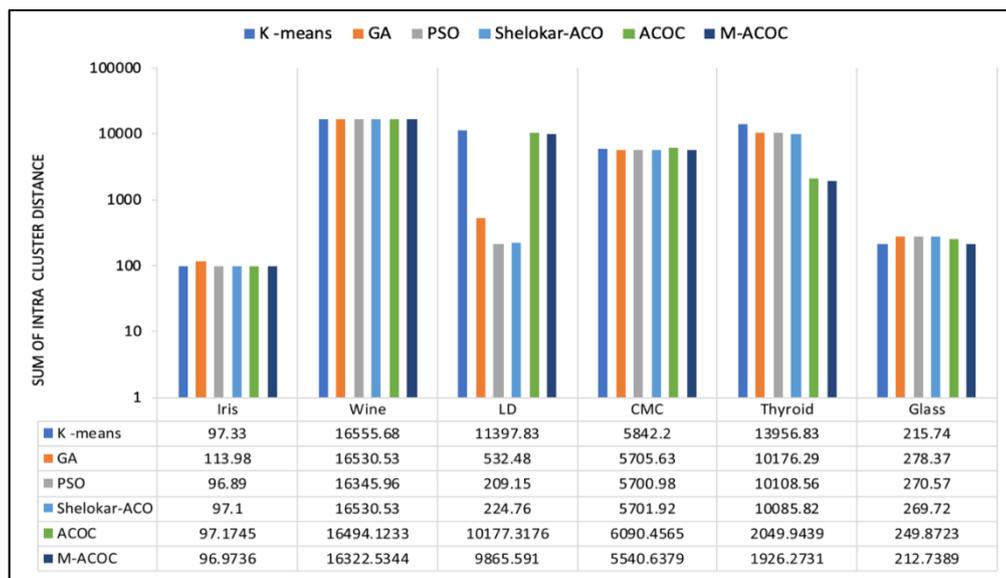


Figure 4. Sum of intra-cluster distances for the six datasets

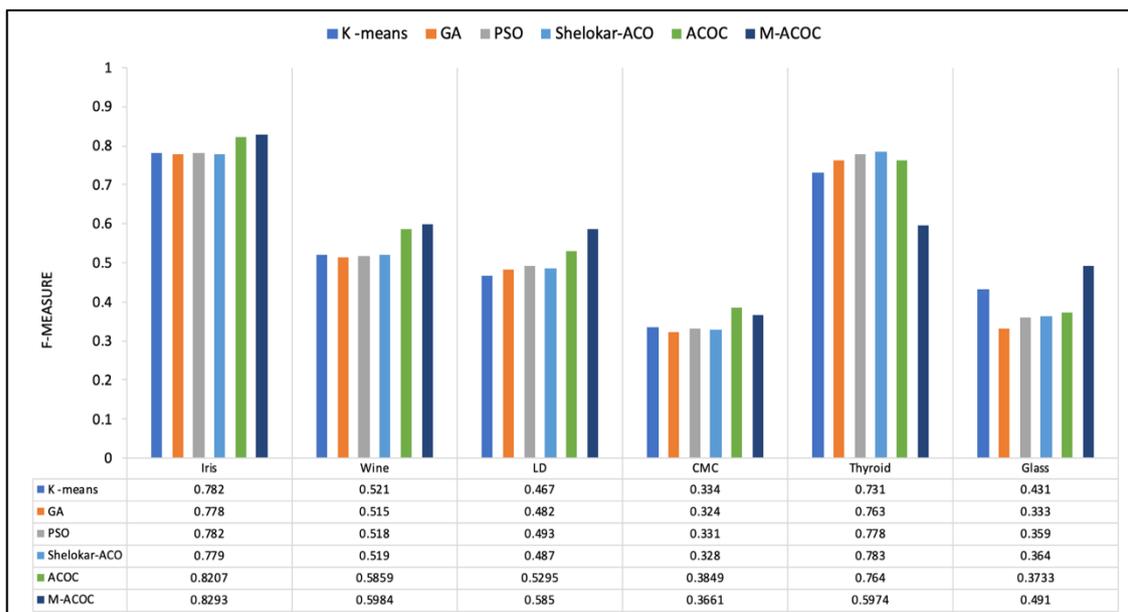


Figure 5. F-measure result for the six datasets

Experiments have also been conducted on three datasets (from the UCI machine learning repository) of different sizes ranging from small to large. Each dataset has a different number of clusters which have the same number of objects. This experiment is to show that the M-ACOC is a centroid-based algorithm works well for clustering when all clusters have similar numbers of objects. These numerical datasets are library, segment and vowel. The library dataset contains 360 objects with 15 classes, each class has 24 objects while the segment dataset contains 2310 objects with seven classes and each class has 330 objects. The third dataset is vowel which contains 990 objects with 11 classes and each class has 90 objects. Figure 6 depicts the results for the sum of intra-class distances where performance comparison was performed on the M-ACOC, ACOC and GA. The lowest intra-class distances were obtained by the M-ACOC.

The best performance was also obtained by the M-ACOC for the F-measure metric when the experiment was, again, performed on the three datasets (refer Figure 7). It can be concluded that the algorithm works well for clustering when all clusters have similar numbers of objects.

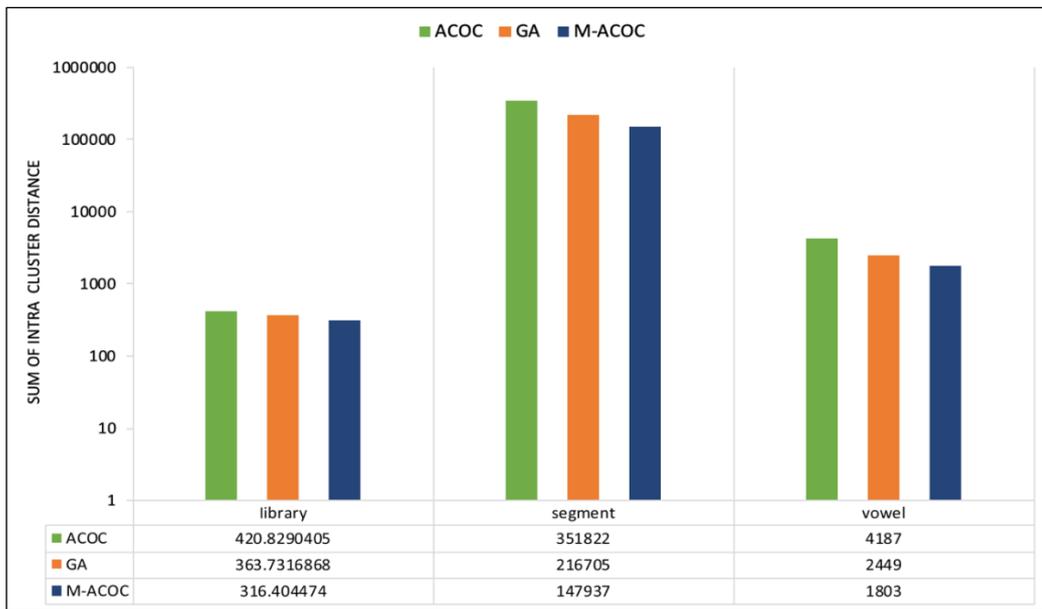


Figure 6. Sum of intra-cluster distances for same cluster size

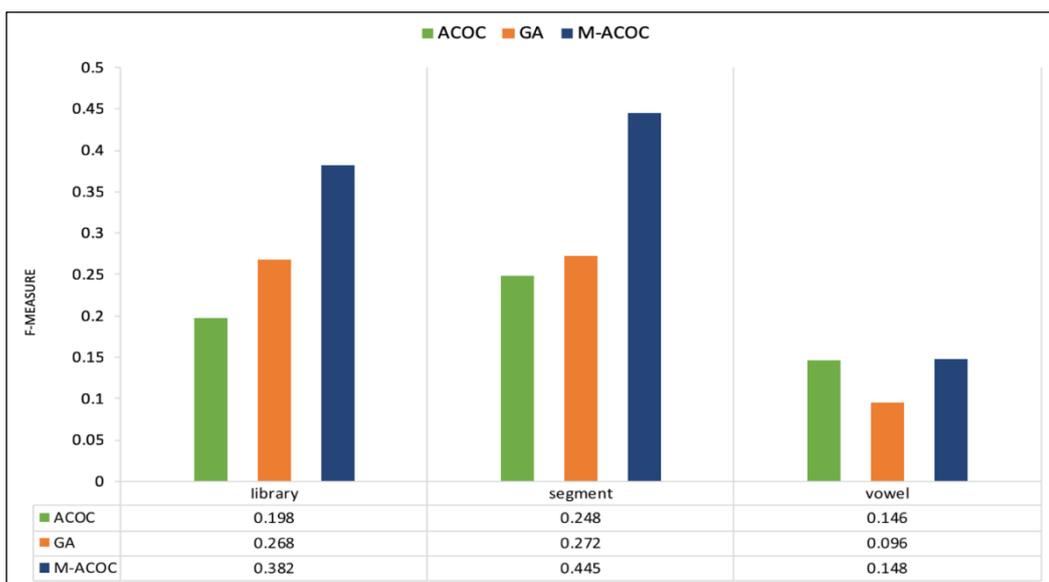


Figure 7. F-measure result for same cluster size

5. CONCLUSION

This study highlights the research on the ACOC algorithm that considers clustering as an approach to solve optimisation problems. It intends to determine the effects of algorithm performance, including slow convergence and the local optimal solution, which are both related to the height exploration process during an algorithm run. This study improves the ACOC algorithm by proposing a new modification rate, *MR*, which is the core engine of the algorithm, by controlling the algorithm convergence. The proposed algorithm, i.e., the M-ACOC, is compared with classical clustering algorithms. The results show that the M-ACOC outperforms classical clustering algorithms in terms of the minimum intra-distance and F-measure results. As a future work, it will be more effective if the algorithm is applied in different datasets using more evaluation criteria to conduct experiments on the behaviour of the algorithm.

ACKNOWLEDGEMENTS

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

REFERENCES

- [1] H. N. K. Al-Behadili, K. R. Ku-Mahamud, and R. Sagban, "Rule pruning techniques in the ant-miner classification algorithm and its variants: A review," in *IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE)*, 2018, pp. 47–56.
- [2] N. M. N. Mathivanan, N. A. Nor, and R. M. Janor, "Improving classification accuracy using clustering technique," *Bull. Electr. Eng. Informatics*, vol. 7, no. 3, pp. 465–470, 2018.
- [3] O. A. M. Jafar and R. Sivakumar, "Ant-based Clustering Algorithms : A Brief Survey," *Int. J. Comput. Theory Eng.*, vol. 2, no. 5, pp. 787–796, 2010.
- [4] D. Xu and Y. Tian, "A Comprehensive Survey of Clustering Algorithms," *Ann. Data Sci.*, vol. 2, no. 2, pp. 165–193, 2015.
- [5] A. M. Jabbar, K. R. Ku-Mahamud, and R. Sagban, "Ant-based sorting and ACO-based clustering approaches: A review," in *IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2018, pp. 217–223.
- [6] S. M. Khatami, M. Maadi, and R. Ramezani, "A Clustering Expert System using Particle Swarm Optimization and K-means++ for Journal Recommendation to Publish the Papers," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 12, no. 2, p. 814, 2019.
- [7] A. Kumar, D. Kumar, and S. K. Jarial, "A review on artificial bee colony algorithms and their applications to data clustering," *Cybern. Inf. Technol.*, vol. 17, no. 3, pp. 3–28, 2017.
- [8] M. Z. Hossain, M. N. Akhtar, R. B. Ahmad, and M. Rahman, "A dynamic K-means clustering for data mining," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 13, no. 2, pp. 521–526, 2019.
- [9] A. N. Alfiyatin, W. F. Mahmudy, and Y. P. Anggodo, "K-Means Clustering and Genetic Algorithm to Solve Vehicle Routing Problem with Time Windows Problem," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 11, no. 2, pp. 462, 2019.
- [10] R. Xu, J. Xu, and D. C. Wunsch, "A Comparison Study of Validity Indices on Swarm-Intelligence-Based Clustering," vol. 42, no. 4, pp. 1243–1256, 2012.
- [11] P. Roy and J. K. Mandal, "An SVD based Real Coded Genetic Algorithm for Graph Clustering," vol. 5, no. 2, pp. 64–71, 2016.
- [12] S. Zhu and L. Xu, "Many-objective fuzzy centroids clustering algorithm for categorical data," *Expert Syst. Appl.*, vol. 96, pp. 230–248, 2018.
- [13] B. Anari, J. Akbari Torkestani, and A. M. Rahmani, "A learning automata-based clustering algorithm using ant swarm intelligence," *Expert Syst.*, no. April, p. e12310, 2018.
- [14] R. Ünlü and P. Xanthopoulos, "Estimating the number of clusters in a dataset via consensus clustering," *Expert Syst. Appl.*, vol. 125, pp. 33–39, 2019.
- [15] S. Karol and V. Mangat, "Evaluation of text document clustering approach based on particle swarm optimization," *Cent. Eur. J. Comput. Sci. Eval.*, vol. 3, no. 2, pp. 69–90, 2013.
- [16] D. P. Sari, D. Rosadi, A. R. Effendie, and D. Danardono, "K-means and bayesian networks to determine building damage levels," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 17, no. 2, p. 719, 2019.
- [17] P. T. T. Binh, T. N. Le, and N. P. Xuan, "Advanced som & k mean method for load curve clustering," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 6, pp. 4829–4835, 2018.
- [18] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Inf. Sci. (Ny)*, vol. 237, pp. 82–117, 2013.
- [19] M. Birattari, "Tuning Metaheuristics: A Machine Learning Perspective," in *Tuning Metaheuristics: A Machine Learning Perspective*, Second edi., vol. 197, Berlin: Springer, 2009, p. 37.
- [20] A. M. Jabbar, "Controlling the Balance of Exploration and Exploitation in ACO Algorithm," *J. Univ. Babylon*, vol. 26, no. 2, pp. 1–9, 2018.
- [21] H. N. K. Al-Behadili, K. R. Ku-Mahamud, and R. Sagban, "Ant colony optimization algorithm for rule-based classification: Issues and potential solutions," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 21, pp. 7139–7150, 2018.

- [22] H. N. K. Al-Behadili, "Intelligent Hypothermia Care System using Ant Colony Optimization for Rules Prediction," *J. Univ. Babylon*, vol. 26, no. 2, pp. 47–56, 2018.
- [23] Y. Kao and K. Cheng, "An ACO-Based Clustering Algorithm," *ANTS Int. Work. Ant Colony Optim. Swarm Intell.*, vol. 4150/2006, pp. 340–347, 2006.
- [24] H. D. Menéndez, F. E. B. Otero, and D. Camacho, "MACOC: A Medoid-Based ACO Clustering Algorithm," in *Swarm Intelligence*, 2014.
- [25] H. D. Menéndez, F. E. B. Otero, and D. Camacho, "Medoid-based clustering using ant colony optimization," *Swarm Intell.*, vol. 10, no. 2, pp. 123–145, 2016.
- [26] C. Koliás, G. Kambourakis, and M. Maragoudakis, "Swarm intelligence in intrusion detection: A survey," *Comput. Secur.*, vol. 30, no. 8, pp. 625–642, 2011.
- [27] W. Shi, T. Weise, P. R. R. Chiong, and B. Catay, "Hybrid PACO with enhanced pheromone initialization for solving the vehicle routing problem with time windows," *Proc. - 2015 IEEE Symp. Ser. Comput. Intell. SSCI 2015*, pp. 1735–1742, 2015.
- [28] R. Sagban, K. R. Ku-Mahamud, and M. S. A. Bakar, "Reactive Max - Min Ant System: An Experimental Analysis of the Combination with K - Opt Local Searches," in *Proceedings of 5th International Conference on Computing and Informatics, ICOCI 2015*, 2015, pp. 300–305.
- [29] U. Boryczka, "Finding groups in data: Cluster analysis with ants," *Appl. Soft Comput. J.*, vol. 9, no. 1, pp. 61–70, 2009.
- [30] K. Bache and M. Lichman, "UCI Machine Learning Repository," *Univ. Calif. Irvine Sch. Inf.*, vol. 20, no. 8, 2013.
- [31] J. O'Rourke and G. T. Toussaint, "Pattern recognition," in *Handbook of Discrete and Computational Geometry, Third Edition*, 2017.
- [32] Y. Kumar and G. Sahoo, "A chaotic charged system search approach for data clustering," *Inform.*, vol. 38, no. 3, pp. 249–261, 2014.

BIOGRAPHIES OF AUTHORS



Ayad Mohammed Jabbar was born in Iraq, Basra, in 1985. He received a Bachelor Certificate in Computer Science from Shatt Al-Arab University College in 2008, and Master of Science (Information Technology) from Universiti Utara Malaysia in 2011. He is currently undertaking a Doctor of Philosophy (Information Technology) at Universiti Utara Malaysia. His current interests include data mining, clustering, artificial intelligence and swarm intelligence.



Ku Ruhana Ku-Mahamud: She holds a Bachelor in Mathematical Sciences and a Master's degree in Computing, both from Bradford University, United Kingdom in 1983 and 1986 respectively. Her PhD in Computer Science was obtained from Universiti Pertanian Malaysia in 1994. As an academic, her research interests include ant colony optimization, pattern classification and vehicle routing problems.



Rafid Sagban holds a Bachelor in Computer Sciences from the University of Babylon in 1999. His Master's degree and PhD in IT are both from the Universiti Utara Malaysia in 2015. Rafid's experience includes over 13 years in academia/industry investigating algorithm analysis, swarm intelligence, and web development.