International Journal on Advanced Science Engineering Information Technology

Vol.11 (2021) No. 3
ISSN: 2088-5334

Improved Self-Adaptive ACS Algorithm to Determine the Optimal Number of Clusters

Ayad Mohammed Jabbar[a,*], Ku Ruhana Ku-Mahamud[b], Rafid Sagban[c]

[a] *Computer Science Department, Shatt Al-Arab University College, Basra, 61001, Iraq*
[b] *School of Computing, Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia*
[c] *Department of Software, University of Babylon, Babylon, 51002, Iraq*
*Corresponding author: *ayadmohammed@sa-uc.edu.iq*

***Abstract*—A fundamental problem in data clustering is how to determine the correct number of clusters. The k-adaptive medoid set ant colony optimization (ACO) clustering (METACOC-K) algorithm is superior in solving clustering problems. However, METACOC-K does not guarantee in finding the best number of clusters. It assumed the number of clusters based on an adaptive parameter strategy that lacks feedback learning. This has restrained the algorithm in producing compact clusters and the optimal number of clusters. In this paper, a self-adaptive ACO clustering (S-ACOC) algorithm is proposed to produce the optimal number of clusters by incorporating a self-adaptive parameter strategy. The S-ACOC algorithm is a centroid-based algorithm that automatically adjusts the number of clusters during the algorithm run. The selection of the number of clusters is based on a construction graph that reflects the influence of a pheromone in algorithm learning. Experiments were conducted on real-world datasets to evaluate the performance of the proposed algorithm. The external evaluation metrics (purity, F-measure, and entropy) were used to compare the results of the proposed algorithm with other swarm clustering algorithms, including a genetic algorithm (GA), particle swarm optimization (PSO), and METACOC-K. Results showed that S-ACOC provides higher purity (50%) and lower entropy (40%) than GA, PSO, and METACOC-K. Experiments were also performed on several predefined clusters, and results demonstrate that the S-ACOC algorithm is superior to GA, PSO, and METACOC-K. Based on the superior performance, S-ACOC can be used to solve clustering problems in various application domains.**

***Keywords*— Data clustering; parameter selection; optimization-based clustering; ant colony optimization.**

Manuscript received 17 Apr. 2020; revised 27 Feb. 2021; accepted 8 Mar. 2021. Date of publication 30 Jun. 2021.

I. INTRODUCTION

Clustering is a data-mining technique that aims to classify data into different groups [1]. This technique organizes unknown data as a set of groups called clusters. The similarity of the items in each cluster is measured based on the extracted features among the data. The extracted features present the relationship and similarities among various types of data [2]. Clustering, as an unsupervised technique in which no class is utilized to predict the result, helps organize different collections of data. Such an approach can be highly beneficial to classifying data. The majority of data worldwide are unlabeled. Thus, the classification technique cannot be applied. The outcome of clusters produced is based on internal criteria, which measure the similarity of members within each cluster and among clusters [3]. Clustering methods can generally be classified into deterministic and optimization approaches. Deterministic approaches employ classical clustering algorithms, whereas optimization approaches regard clustering as an optimization problem. Similarity can be measured according to the clustering problem's target, such as using the minimum squared error for centroid and medoid or the density of members in each cluster or as a tree in hierarchical clustering. This scenario affects structural clusters, the specification of clusters, and the number of clusters produced, known as *k*. The number of clusters in unsupervised learning, which is often considered a parameter required to be optimized [4], is a vital hyperparameter for most clustering algorithms and generally unknown. Accurately determining the parameter value is a crucial task when the only information available is the numerical values of different features. Despite the existence of diverse algorithmic approaches for solving this problem, no de facto optimal approach is available [4]. Researchers use different objective functions (validity indices) as guidelines for optimal partitions. The validity indices (CVIs) are a relative evaluation metric that considers cohesion and separation

criteria to determine the optimal partitions reflecting the optimal number of clusters. Algorithms that utilize those CVIs as an objective function are normally called automatic clustering algorithms [5]–[7]. Several studies have utilized the particle swarm optimization (PSO) algorithm to generate the number of clusters using a validity index as an objective function with different conditions to improve the clustering results [8]–[11]. Such conditions have resulted in obtaining the best centroids for each cluster. The clusters are then examined to provide the best solutions. However, these studies did not consider any relationship among the obtained clusters, although the results have been improved by finding enhanced local centroids located in the best centroids [9]. Similar research has adopted a single validity index as an objective function of the artificial bee colony algorithm to determine the optimal number of clusters [12]. A single validity index has also been used in PSO algorithm whereby each particle will generate a clustering solution based on randomly generated centroids [13].

In this particular study, an encoding scheme based on the random number of clusters generated in each encoding and a threshold that was initialized as the static value will control the active clusters. Other studies have employed hybrid algorithms, merging the advantages of multi-objective PSO and simulated annealing for automated clustering. In particular, three validity indices are simultaneously optimized as a single-objective function to produce a suitable number of clusters [14]. Related research uses a hybrid algorithm that combines differential equations and fuzzy *c*-means for clustering. The algorithm utilizes a self-adaptive approach to the trade-off between exploration and exploitation. The mechanism is based on measuring the gap between two clustering solutions that reflect the diversity of the solutions. The value of the self-adaptive algorithm is used to identify the state of the search to move either towards exploration or actual exploration [15]. In 2017, the grey wolf optimizer was introduced as an adaptive algorithm for image segmentation [16]. The algorithm utilizes the Davies–Bouldin index as the objective function to determine the number of clusters. Nevertheless, the algorithm has only been applied in image segmentation, and the number of clusters is determined using an adaptive strategy.

An important question in the process of solving the clustering problem is whether the current memory can handle different cluster numbers within each iteration. The answer is that due to the difficulty of memory management when different clusters exist in the same memory, the information used in the clustering assignment is related to several regain attempts on the search space and represents a diverse number of clusters. Management of the memory model is an essential issue for all algorithms and is considered the key that keeps the search process controlled with the trade-off between exploration and exploitation for optimal solutions [17]–[19]. Ant colony optimization (ACO) is one of the best algorithms employed in different application domains, such as classification, feature selection, and clustering [20]–[25]. In contrast to other algorithms, ACO uses an adaptive memory for its problem that requires to be optimized, and it is important for its performance in terms of achieving better results during the algorithm run [26]. The search regions are recorded in the adaptive memory of the algorithm to find the regions with the best clustering solutions, which has improved during the algorithm run. ACO is the only algorithm that responds to transferring the currently recorded search regions to future iterations to be used and improved for better solutions [27], [28]. The K-adaptive MEdoid seT ACO clustering (METACOC-K) algorithm is a medoid-based algorithm that follows the same framework as ACO for clustering problems [29]. This algorithm follows the same idea used in adaptive approach parameter selection to determine the number of clusters, *k*. However, the adaptive strategy utilized in the algorithm is based on a random assumption with no positive feedback representing the quality of *k*. The search process does not guide the algorithm, and *each ant randomly generates k value* as a prior step before performing the clustering assignment.

For the parameter optimization problem, subject to solving the number of clusters, *k*, three approaches have been considered; they are adaptive, used by the METACOC-K algorithm; self-adaptive; and search-adaptive [30]–[32]. The self-adaptive approach employs the search space of the algorithm by adding parameter *k* into the search space. Hence, the algorithm optimizes *k* within its algorithm graph and during clustering optimization. The best value of *k* is optimized under the feedback of the clustering assignment. The search-adaptive approach utilizes an external algorithm that independently runs to optimize the parameter. The external algorithm employs its graph designed for *k*. The self-adaptive approach is considered a better approach to use in this research, given only one parameter that does not require running other external algorithms. The run time required by the self-adaptive approach is less than the time spent by the external algorithm. The feedback of the self-adaptive approach is sufficient to identify the best value of *k*.

This study proposes adding a self-adaptive approach that guides the search process for optimal k value selection and intensifies the search for the selected *k* value to find optimal clustering assignments. The feedback of each *k* value is important to direct the search process for selecting the optimal *k* value. This study extends the existing memory by adding a new search space entry representing all k values in the search space of ACO for data clustering (ACOC) algorithm, a centroid-based algorithm [33]. The new search space will be updated under the quality of the clustering assignment. The proposed self-adaptive ACOC (S-ACOC) approach will enhance the ACOC algorithm to optimize *k* value and improve clustering.

II. Materials and Method

The proposed self-adaptive approach for selecting the optimal number of clusters is integrated into the ACOC algorithm, the original algorithm used in clustering [33]. The only difference between METACOC-K and ACOC is that the former is medoid-based, whereas the latter is centroid-based. ACOC is a static algorithm that requires the number of clusters as input from the user. In this study, the ACOC algorithm has been used as the base algorithm because a centroid-based algorithm has a wide search space containing many cluster centers that must be optimized. This is in contrast to the medoid-based algorithm, where the number of cluster centers depends only on the number of instances. The research is conducted in three (3) stages; parameter tuning,

multi-memory assignment, and performance evaluation as shown in Fig 1.

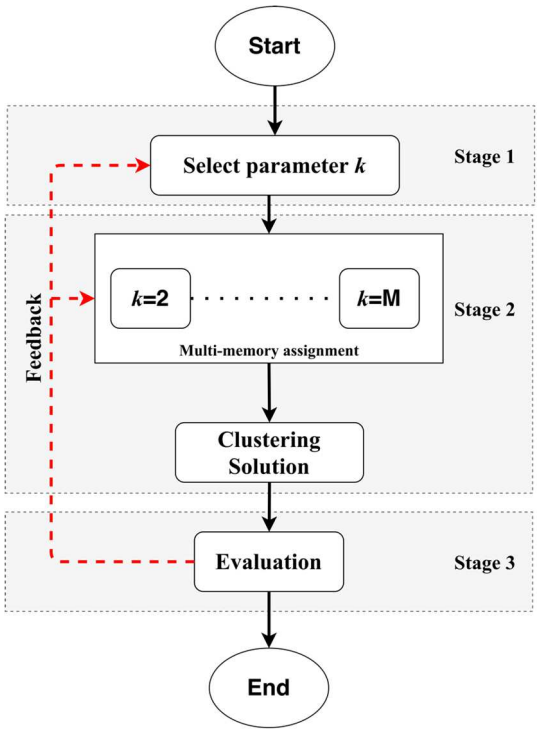

Fig. 1 The flowchart of the S-ACO algorithm

In the first stage, an optimal value will be obtained from several values and assigned to a k, representing the number of clusters. The process of defining several memory spaces (multi-memory) in which each memory will store the information regarding only a parameter is performed. Each memory contains the best centroids obtained during the run and represents the best clustering assignment based on the information stored. This information is the amount of pheromone laid by the ants denoted here as the feedback represents the quality of the solution. The output of this stage is a clustering solution, which is evaluated in the third stage. In the evaluation process (third stage), the measurement of external metrics such as the purity, F-measure, and entropy are calculated. These are the common performance metrics used in the classification and clustering domains [34]. Note here the search space for the parameter, k, is in the range of [2, M], where M is a predefined parameter that represents the maximum number of clusters set by the user. The benchmark datasets used in the evaluation are from different application domains obtained from the University of California, Irvine (UCI) repository. Comparisons of the proposed algorithm were made with the genetic algorithm (GA), PSO, and METACOC-K algorithm.

This study defines a new search space that contains all possible k values denoted as parameter memory that must be optimized, whereas the search space that contains the clustering assignment is denoted as assignment memory (multi-memory), as shown in Fig. 2. The proposed S-ACOC forces the ants to select a value for k from the parameter memory. It then performs clustering in accordance with the selected value. The value of k is selected in accordance with the amount of pheromone available on the nodes, in which the node with a high amount of pheromone has a high probability of being selected by ants. Each ant first selects a value for k and then generates random centroids. The number of centroids is equal to k. The generated centroids are used to perform a clustering assignment. This modification enables the algorithm to store the information about the clustering assignment separately in each assignment memory. Thus, the information representing the assignment of two clusters does not affect the information representing the remaining clusters [35].

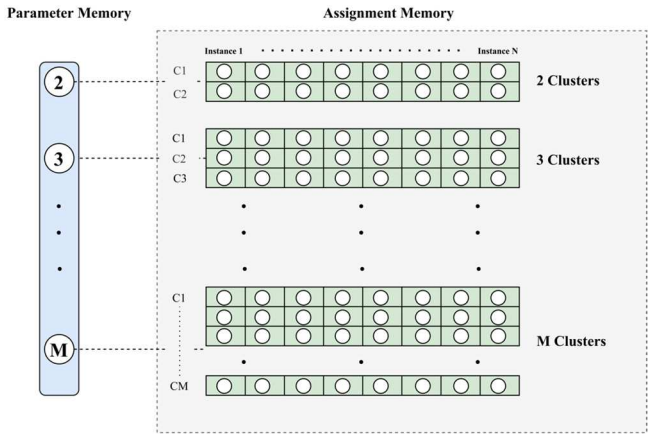

Fig. 2 Self-adaptive approach of S-ACOC

Data clustering can be described as an optimization problem that aims to optimize an objective function to minimize or maximize a predefined objective value. However, the objective of any clustering algorithm is to find the optimal configuration of groups generated by the algorithm. The mathematical formulation and computing procedure of S-ACOC are presented as follows. Given x instances, S-ACOC automatically produces k clusters, and each cluster has dissimilar compact instances of other clusters. Each ant performs clustering by assigning each instance to one unique cluster center, as shown in Equation (1). The instance that has a short distance with a cluster center is marked as an instance that belongs to that cluster center. Equation (1) uses only a pre-objective function to assign each instance to a close centroid because it cannot use an objective function to evaluate the overall clustering assignment. We need an additional post-objective function to evaluate the clustering assignment generated from Equation (1). Note that n is the total number of instances; n_i is the total number of instances belonging to the ith cluster; C_i is the centroid of the ith cluster; and C is the global centroid [36].

$$Minimize\ Intra = \sum_{i=1}^{k} \sum_{j=1}^{n_i} d(x_{ij}, C_i) \quad (1)$$

where

$$d(x_{ij}, C_i) = \sqrt{\|x_{ij} - C_i\|^2} \quad (2)$$

Equation (3) is the post-objective function that maximizes a criterion to represent the optimal clustering assignment. The objective function used in this research is the Calinski–Harabasz (CH) metric, which is an internal metric evaluating the optimal number of clusters [37]. The CH value is the objective function (Q) that guides the search process in the optimal clustering assignment.

$$Maximize\ CH(k) = \frac{B_c(k)}{(k-1)} \Big/ \frac{W_c\,(k)}{(n-1)} \quad (3)$$

where W_c is the within-cluster sum of the secured error (Equation (4)), and B_c is the between-cluster sum of the secured error (Equation (5)) [38].

$$B_c(k) = \sum_{i=1}^{k} n_i \|C_i - C\|^2 \quad (4)$$

$$W_c(k) = \sum_{i=1}^{k} \sum_{j=1}^{n_i} \|x_{ij} - C_i\|^2 \quad (5)$$

In S-ACOC, the clustering solution is modeled as a graph of nodes connected to each instance, and each instance has a number of nodes that are equal to k, as shown in Fig. 3. The graph is represented as a matrix of n by k, where n is the number of instances. Each ant will travel from node to node sequentially. Fig. 3 represents a clustering assignment of three clusters built by ants, which contains clustering strings of (1, 2, 2, 3, 2, 3); each instance is assigned to the ith cluster. The clustering solution is performed using pheromone intensity and heuristic information. Each ant should first select k value from the parameter memory to generate random centroids for building clustering assignments. The final clustering solution can be represented as (3, 1, 2, 2, 3, 2, 3), where the first digit indicates $k = 3$.

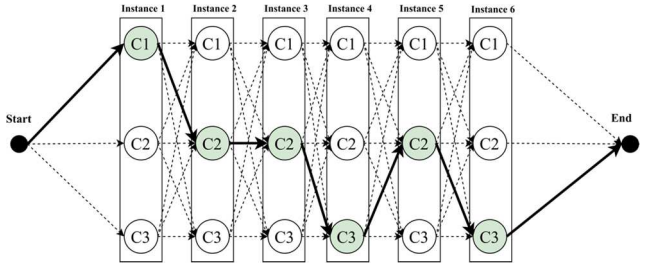

Fig. 3 Construction graph for S-ACOC

In the graph, ants select a value for k on the basis of Equation (6), in which only pheromone information is available in making the decision. Once the ants select an appropriate k value, the assignment is performed until all instances are grouped when memory list tb^k is full, and the ants construct a solution.

$$p_p = \begin{cases} \tau_{p_s}^{\alpha}, & if\ q < q_0; \\ S, & otherwise, \end{cases} \quad (6)$$

where S is calculated as

$$S = \frac{\tau_{p_s}^{\alpha}}{\sum_{s=2}^{k_{max}} \tau_{p_s}^{\alpha}} \quad (7)$$

In Equation (6), q is a random value in the range [0–1], and q_0 is a predefined parameter value in the range [0–1]. If $q > q_0$, then the search is explorative. The pheromone update in S-ACOC deposits one node j of instance x to be considerably attractive in advance iterations. The amount of pheromone that should be deposited is calculated as

$$\tau_{xj,p_s} \leftarrow (1 - \rho)\tau_{xj,p_s} + \left(Q_{if\ xj\ and\ p_s \in clustering\ solution}\right) \quad (8)$$

where Q is the value of the objective function (Equation (3)).

The complete procedure of S-ACOC, consisting of seven steps, is as follows:

Step 1 Initialization: The pheromone memory (PM) and parameter memory (P_aM) are set to τ_0, i.e., a small value of 0.1.

Step 2 Initialization of all ants: A new iteration is started with all ants through an empty clustering solution, and each ant generates random centroids.

Step 3 Parameter selection: An ant selects a parameter value (k value) from the parameter memory (P_aM) on the basis of Equation (6).

Step 4 Selection of an instance: Each ant performs a clustering process with the assignment of all instances to centroids in sequence. The centroids are obtained randomly in Step 2.

Step 5 Selection of the best ant: The best ant in the current iteration is called the best iteration solution, which produces the highest value of CH as a post-objective function by using Equation (3). The value of CH indicates the current optimal k in the current iteration. The iteration-best solution is compared with the best-so-far solution that represents the best clustering solution of the overall iteration, and the better one will be the new best-so-far solution.

Step 6 Updating pheromone trails: Only the best ant in each iteration is allowed to update the pheromone for its clustering solution for both memories, namely, PM and P_aM, by using Equation (8). Pheromone update is performed on the nodes of PM, and the best ant selects the current k in the current iteration.

Step 7 Checking the termination condition: The algorithm stops when the number of iterations exceeds the prescribed limit, and the best-so-far solution is printed. Otherwise, Step 2 is repeated.

III. Results and Discussion

The performance evaluation of S-ACOC is conducted using external evaluation criteria, including purity, F-measure, and entropy. The purpose of using such criteria for performance evaluation is that the existing clustering algorithm uses different objective functions that are calculated as the clustering solution on the basis of internal evaluation criteria, in which each objective function can provide diverse clustering results [39]. Using external evaluation is effective because the criteria are supervised approaches that calculate the clustering solution on the basis of trueness knowledge. The purity and F-measure evaluation criteria consider a clustering solution to be optimal if maximized. Conversely, the entropy evaluation criterion regards a clustering solution as optimal if minimized. Purity is the percentage of the total number of instances that are correctly categorized, as shown in Equation (9), where $\Omega = w1, w2, \ldots, wk$ is the set of clusters, and $\Gamma = c1, c2, \ldots, cj$ is the set of classes [40].

$$\text{Purity}(\Omega, \Gamma) = \frac{1}{N} \sum_k maxj \left| w_i \cap c_j \right| \quad (9)$$

The calculation of F-measure is based on the measurement of precision and recall. Precision and recall are calculated using Equations (10) and (11), respectively.

$$Precision = \frac{TP}{TP+FP} \quad (10)$$

$$Recall = \frac{TP}{TP+FN} \quad (11)$$

where TP is the true positive, FP is the false positive, and FN is the false negative. F-measure can then be calculated as

$$\mathrm{F-Measure} = \frac{(\beta^2+1)Precision*Recall}{\beta^2 Precision+Recall} \quad (12)$$

where β is a constant that we can use to penalize false negatives more strongly than false positives by selecting its value >=1, thereby providing further weight on recall.

The calculation of entropy is shown in Equation (13), which measures the entropy for single clustering w [41]. The total entropy of the clustering is calculated in Equation (14).

$$H(w) = -\sum_{c\in C} P(w_c) log_2 P(w_c) \quad (13)$$

$$H(\Omega) = \sum_{w\in\Omega} H(w) \frac{N_w}{N} \quad (14)$$

The benchmark datasets are supervised benchmarks used in clustering and classification tasks. The benefit of using a supervised benchmark is that a clustering algorithm can be evaluated using external evaluation metrics, such as F-measure and entropy, in which both metrics use the dataset label for the evaluation. Ten benchmark datasets are extracted from the UCI Machine Learning Repository [42]. The datasets belong to different problem areas, such as disease, image, and analysis, and they differ in dimension and size. They are classified based on the number of features, such as small, medium, large, and very large [43]. Seven datasets, namely, breast cancer, breast tissue, *E. coli*, Haberman, hepatitis, iris, and wine, are classified as small datasets. The ionosphere is classified as a medium dataset, and Libras and sonar are considered large datasets. Table I depicts the characteristics of the datasets.

TABLE I
DESCRIPTION OF UCI DATASETS USED IN THE EXPERIMENTS

Name	Feature Size	Class	Instance
Breast cancer	9	2	699
Breast tissue	9	6	106
E. coli	7	6	336
Libras	90	15	360
Haberman	3	2	306
Hepatitis	19	2	155
Ionosphere	34	2	351
Sonar	60	2	208
Iris	4	3	150
Wine	13	3	178

The evaluation of the proposed algorithm is performed against state-of-the-art clustering algorithms, including GA, METACOC-K, and PSO [44]. GA and PSO use DB as an objective function, whereas METACOC-K uses a silhouette index as an objective function. The proposed algorithm, S-ACOC, uses CH as an objective function.

To evaluate the performance of the proposed algorithm and determine if the comparison among algorithms is fair, this study sets the algorithm parameters in accordance with the values used in the literature of clustering algorithms. The setting of the algorithms is listed in Table II, which shows all clustering algorithms, namely, GA, PSO, METACOC-K, and S-ACOC.

TABLE II
VALUES OF THE PARAMETERS OF THE ALGORITHMS

GA	PSO	METACOC-K / S-ACOC
Population = 50	Population = 50	Ants = 50
Crossover = 0.8	Inertia Weight = 1	Probability threshold = 0.001
Mutation rate = 0.001	Inertia Weight Damping Ratio = 0.00	Evaporation rate = 0.001
Iterations = 1000	Iterations = 1000	Iterations = 1000
Maximum number of clusters = 20	Global learning coefficient = 2.0	
	Personal learning coefficient =1.5	
	Maximum number of clusters= 20	

The results of the experiments for evaluating the performance of the S-ACOC algorithm, which is compared with GA, PSO, and METACOC-K, are shown in Tables III, IV, V, and VI. The comparisons are performed based on i) external evaluation metrics, i.e., purity, F-measure, and entropy, and ii) the number of clusters. Table III depicts the results concerning the purity metric, which indicate that S-ACOC outperforms the other algorithms in five datasets (50%), whereas METACOC-K obtains the best results in only three datasets (30%). GA and PSO obtain the best result in only one dataset (10%).

TABLE III
AVERAGE PURITY RESULTS OF THE CLUSTERING ALGORITHMS

Dataset	GA	PSO	METACOC-K	S-ACOC
Breast cancer	0.938	0.951	0.951	**0.957**
Breast tissue	0.327	0.331	**0.389**	0.377
E. coli	0.607	0.630	**0.755**	0.748
Libras	0.152	0.156	**0.403**	0.071
Haberman	0.744	0.748	0.737	**0.752**
Hepatitis	0.797	0.797	0.793	**0.798**
Ionosphere	0.723	**0.741**	0.647	0.707
Sonar	**0.581**	0.558	0.538	0.552
Iris	0.729	0.786	0.667	**0.893**
Wine	0.676	0.703	0.646	**0.726**

Comparison between S-ACOC and GA indicates that the S-ACOC algorithm produces the best results in seven datasets (70%), namely, breast cancer, breast tissue, *E. coli*, Haberman, hepatitis, iris, and wine, whereas GA produces the best results in only three datasets, namely, Libras, ionosphere, and sonar. Comparison between S-ACOC and PSO indicates that the S-ACOC algorithm produces the best results in seven datasets (70%), namely, breast cancer, breast tissue, *E. coli*, Haberman, hepatitis, iris, and wine, whereas PSO obtains the best results in only three datasets, namely, Libras, ionosphere, and sonar (30%). Comparison between S-ACOC and METACOC-K indicates that the S-ACOC algorithm produces the best results in seven datasets (70%), namely, breast cancer, Haberman, hepatitis, ionosphere, sonar, iris, and wine, whereas

METACOC-K produces the best results in only three datasets, namely, breast tissue, *E. coli*, and Libras (30%).

The results of the second comparison by using the F-measure metric are shown in Table IV. In this case, the best clustering is produced if the F-measure value is maximized. Overall, METACOC-K produces the best result in six datasets (60%), whereas S-ACOC and GA produce the best results in two datasets (20%). The PSO algorithm does not produce any best result.

TABLE IV
AVERAGE F-MEASURE RESULTS OF THE CLUSTERING ALGORITHMS

Dataset	GA	PSO	METACOC-K	S-ACOC
Breast cancer	0.896	0.840	**0.948**	0.925
Breast tissue	0.187	0.210	0.222	**0.231**
E. coli	0.514	0.575	**0.786**	0.739
Libras	0.070	0.073	**0.273**	0.017
Haberman	**0.548**	0.528	0.273	0.411
Hepatitis	0.680	0.680	**0.744**	0.740
Ionosphere	0.531	0.419	**0.799**	0.601
Sonar	0.409	0.499	**0.650**	0.503
Iris	0.622	0.706	0.568	**0.821**
Wine	**0.509**	0.492	0.504	0.271

Comparison between S-ACOC and GA indicates that S-ACOC performs better than GA. In particular, the former produces the best results in eight datasets, namely, breast cancer, breast tissue, *E. coli*, hepatitis, ionosphere, sonar, iris, and wine (80%), whereas the latter produces the best results in only two datasets, namely, Libras and Haberman (20%). Comparison between S-ACOC and PSO demonstrates that S-ACOC outperforms PSO in seven datasets, namely, breast cancer, breast tissue, *E. coli*, hepatitis, ionosphere, sonar, and iris (70%), whereas PSO produces the best results in only three datasets, namely, Libras, Haberman, and wine (30%). In the comparison between S-ACOC and METACOC-K, METACOC-K outperforms S-ACOC in producing an average F-measure in 70% of the datasets, including breast cancer, *E. coli*, Libras, hepatitis, ionosphere, sonar, and wine, whereas S-ACOC produces the best results in only three datasets, namely, breast tissue, Haberman, and iris.

The third comparison results, which is on entropy, are shown in Table V. The best clustering result is obtained if the entropy value is minimized. Based on the results, S-ACOC outperforms GA, PSO, and METACOC-K in four datasets (40%), whereas the other algorithms produce the best results in only two datasets (20%).

TABLE V
AVERAGE ENTROPY RESULTS OF THE CLUSTERING ALGORITHMS

Dataset	GA	PSO	METACOC-K	S-ACOC
Breast cancer	0.329	0.224	**0.174**	0.257
Breast tissue	1.554	1.398	2.047	**0.661**
E. coli	0.828	0.742	0.738	**0.575**
Libras	1.121	1.009	1.675	**0.520**
Haberman	0.813	0.801	0.831	**0.795**
Hepatitis	**0.716**	**0.716**	0.730	0.734
Ionosphere	0.810	**0.781**	0.933	0.816
Sonar	**0.958**	0.969	0.992	0.988
Iris	0.489	0.422	**0.333**	0.394
Wine	0.726	0.812	**0.654**	0.749

Comparison between S-ACOC and GA indicates that S-ACOC produces the best results in six datasets, namely, breast cancer, breast tissue, *E. coli*, Libras, Haberman, and iris (60%), whereas GA produces the best results in only four datasets, namely, hepatitis, ionosphere, sonar, and wine (40%). S-ACOC outperforms PSO in six datasets, namely, breast tissue, *E. coli*, Libras, Haberman, iris, and wine (approximately 60%), whereas PSO produces the best results in only four datasets (40%), namely, breast cancer, hepatitis, ionosphere, and sonar. Comparison between S-ACOC and METACOC-K shows that S-ACOC performs better than METACOC-K. Specifically, S-ACOC produces the best results in seven datasets, namely, breast cancer, breast tissue, *E. coli*, Libras, Haberman, ionosphere, and sonar (70%), whereas METACOC-K produces the best results in only three datasets, namely, hepatitis, iris, and wine (30%).

The final comparison is on the number of clusters produced by each algorithm. The number of clusters is part of the evaluation on which algorithm can produce the optimal k for each dataset with different density regions. Table VI shows the average number of clusters. The S-ACOC algorithm performs better than GA and PSO, whereas it is almost comparable with METACOC-K. S-ACOC produces the best number of clusters in six datasets, whereas METACOC-K produces the best results in five datasets. GA produces the best results in only two datasets, and PSO produces optimal results in only one dataset.

TABLE VI
AVERAGE NUMBER OF CLUSTERS PRODUCED BY THE ALGORITHMS

Dataset	GA	PSO	METACOC-K	S-ACOC	Actual number
Breast cancer	**2**	11	**2**	**2**	2
Breast tissue	5	4	9	2	6
E. coli	3	3	4	3	6
Libras	5	5	16	2	15
Haberman	4	5	**2**	**2**	2
Hepatitis	6	6	**2**	**2**	2
Ionosphere	4	7	**2**	**2**	2
Sonar	5	5	**2**	**2**	2
Iris	**3**	**3**	2	**3**	3
Wine	**3**	5	2	11	3

Analysis of the results in relation to the size of datasets reveals that the S-ACOC algorithm can produce the best results on small-sized datasets. The algorithm is based on centroids; thus, exploring among small numbers of attributes to find the optimal centroids is easier. Furthermore, the adaptive strategy with multiple centroids forces the algorithm to intensify the search space in finding the best clustering assignment.

The comparisons in Tables III, IV, and V are summarized in Fig. 4, which shows that S-ACOC obtains the best results in purity and entropy. METACOC-K obtains better results than S-ACOC only in F-measure. Hence, in general, S-ACOC can obtain better clustering results than GA, PSO, and METACOC-K.

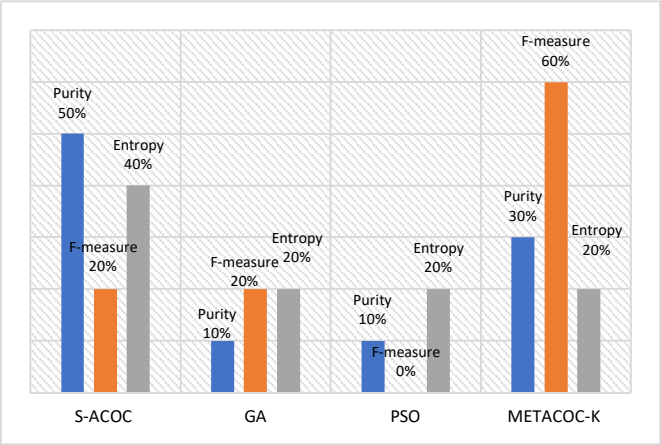

Fig. 4 Quality metrics for clustering (external metrics)

IV. Conclusion

This study addresses the problem of improving a clustering solution through ACO by determining the best number of clusters via a self-adaptive strategy instead of the existing adaptive strategy used in ACO. The improvement in the solution is achieved, which proves that the number of clusters can be adjusted during algorithm run based on the feedback that represents the quality of the current number of clusters without losing the history of the search space of the best assignment. Different number of clusters are examined in the early search stage to explore the quality of solutions through time and move the search process to the end of the search stage. Promising regions with enhanced quality of solutions can then be determined based on the best number of clusters obtained. Therefore, the result of the performance evaluation by using external metrics, namely, purity, F-measure, and entropy, shows that the proposed algorithm outperforms other swarm clustering algorithms. In particular, the S-ACOC algorithm produces higher purity (50%) and lower entropy (40%) than GA, PSO, and METACOC-K. In terms of the number of clusters, S-ACOC succeeds in finding the exact number of clusters (approximately 60%) of the datasets.

Although the S-ACOC algorithm provides better clustering results in terms of external evaluation criteria and the number of clusters produced, it still has a limitation in its assignment memory based on centroid assignments. S-ACOC has multi-memory representing different numbers of assignment, but the amount of pheromone laid in each memory may represent diverse assignments related to the same number of clusters. The current centroid value changes from one iteration to another because of the changes resulting from the random selection process. The amount of pheromone is worthless, and the accumulated pheromone does not reflect the accurate clustering assignment. Future research should focus on improving centroid selection by developing a stochastic centroid memory that ignores the random selection process for centroid selection. The proposed algorithm can also be applied to datasets that belong to different application problems, such as text, audio, and video. Other evaluation criteria and different objective functions can be used to find a highly accurate number of clusters.

Acknowledgment

The Ministry of Higher Education Malaysia has funded this study under the Transdisciplinary Research Grant Scheme, TRGS/1/2018/UUM/02/3/3 (S/O code 14163).

References

[1] T. Herawan, R. Ghazali, and M. M. Deris, "A New Algorithm for Incremental Web Page Clustering Based on k-Means and Ant Colony Optimization," *Adv. Intell. Syst. Comput.*, vol. 287, pp. 347–357, 2014, doi: 10.1007/978-3-319-07692-8.

[2] A. M. Jabbar, K. R. Ku-Mahamud, and R. Sagban, "An improved ACS algorithm for data clustering," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 17, no. 3, pp. 1506–1515, 2019, doi: 10.11591/ijeecs.v17.i3.pp1506-1515.

[3] K. Aparna, "Evolutionary computing based hybrid bisecting clustering algorithm for multidimensional data," *Sādhanā*, vol. 0123456789, 2019, doi: 10.1007/s12046-018-1011-y.

[4] 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, doi: 10.1016/j.eswa.2019.01.074.

[5] A. Mutoh, M. Wada, and K. Amano, "Comprehensive cluster validity Index based on structural simplicity," pp. 1–2, 2019.

[6] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona, "An extensive comparative study of cluster validity indices," *Pattern Recognit.*, vol. 46, no. 1, pp. 243–256, 2013.

[7] R. Xu, J. Xu, and D. C. Wunsch, "A Comparison Study of Validity Indices on Swarm-Intelligence-Based Clustering," *IEEE Trans. Syst. Man. Cybern.*, vol. 42, no. 4, pp. 1243–1256, 2012.

[8] S. Zhou and Z. Xu, "A novel internal validity index based on the cluster centre and the nearest neighbour cluster," *Appl. Soft Comput. J.*, vol. 71, pp. 78–88, 2018, doi: 10.1016/j.asoc.2018.06.033.

[9] C. W. Wang and J. I. G. Hwang, "Automatic clustering using particle swarm optimization with various validity indices," *2012 5th Int. Conf. Biomed. Eng. Informatics, BMEI 2012*, no. Bmei, pp. 1557–1561, 2012, doi: 10.1109/BMEI.2012.6513143.

[10] X. Gao and S. Wu, "CUBOS: An Internal Cluster Validity Index for Categorical Data," *Teh. Vjesn. - Tech. Gaz.*, vol. 26, no. 2, pp. 486–494, 2019, doi: 10.17559/tv-20190109015453.

[11] M. Ren, P. Liu, Z. Wang, and J. Yi, " A Self-Adaptive Fuzzy c -Means Algorithm for Determining the Optimal Number of Clusters ," *Comput. Intell. Neurosci.*, vol. 2016, no. 1, pp. 1–12, 2016, doi: 10.1155/2016/2647389.

[12] R. J. K. F. E. Zulvia, "Automatic clustering using an improved artificial bee colony optimization for customer segmentation," *Knowl. Inf. Syst.*, no. 43, 2018, doi: 10.1007/s10115-018-1162-5.

[13] R. J. Kuo and F. E. Zulvia, "Automatic Clustering Using an Improved Particle Swarm Optimization," *J. Ind. Intell. Inf.*, vol. 1, no. 1, pp. 46–51, 2013.

[14] A. Abubaker, A. Baharum, and M. Alrefaei, "Automatic clustering using multi-objective particle swarm and simulated annealing," *PLoS One*, vol. 10, no. 7, 2015, doi: 10.1371/journal.pone.0130995.

[15] S. Supratid and P. Julrode, "Differential Evolution for Fuzzy Clustering Using Self-Adaptive Trade-Off Between Exploitation and Exploration," *Res. J. Appl. Sci.*, pp. 452–460, 2014.

[16] S. Kapoor, I. Zeya, C. Singhal, and S. J. Nanda, "A Grey Wolf Optimizer Based Automatic Clustering Algorithm for Satellite Image Segmentation," *Procedia Comput. Sci.*, vol. 115, pp. 415–422, 2017, doi: 10.1016/j.procs.2017.09.100.

[17] 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)*, Apr. 2018, pp. 217–223.

[18] A. M. Jabbar, K. R. Ku-Mahamud, and R. Sagban, "Balancing Exploration and Exploitation in ACS Algorithms for Data Clustering," vol. 97, no. 16, pp. 4320–4333, 2019.

[19] M. López-Ibáñez and T. Stützle, "An experimental analysis of design choices of multi-objective ant colony optimization algorithms," *Swarm Intell.*, vol. 6, no. 3, pp. 207–232, 2012, doi: 10.1007/s11721-012-0070-7.

[20] H. N. K. AL-Behadili, K. R. Ku-Mahamud, and R. Sagban, "Hybrid ant colony optimization and genetic algorithm for rule induction," *J. Comput. Sci.*, vol. 16, no. 7, pp. 1019–1028, 2020, doi: 10.3844/JCSSP.2020.1019.1028.

[21] H. N. K. Al-Behadili, R. Sagban, and K. R. Ku-Mahamud, "Adaptive parameter control strategy for ant-miner classification algorithm," *Indones. J. Electr. Eng. Informatics*, vol. 8, no. 1, pp. 149–162, 2020, doi: 10.11591/ijeei.v8i1.1423.

[22] 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, doi: 10.1109/ISCAIE.2018.8405448.

[23] H. N. K. Al-behadili, K. R. Ku-mahamud, and R. Sagban, "Annealing strategy for an enhance rule pruning technique in ACO-based rule classification," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 16, no. 3, pp. 1499–1507, 2019, doi: 10.11591/ijeecs.v16.i3.pp1499-1507.

[24] 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.

[25] H. N. K. Al-Behadili, K. R. Ku-Mahamud, and R. Sagban, "Hybrid Ant Colony Optimization and Iterated Local Search for Rules-Based Classification," *J. Theor. Appl. Inf. Technol.*, vol. 98, no. 04, pp. 657–671, 2020.

[26] T. İnkaya, S. Kayalıgil, and N. E. Özdemirel, "Ant Colony Optimization based clustering methodology," *Appl. Soft Comput.*, pp. 301–311, 2015.

[27] A. M. Jabbar, K. R. Ku-Mahamud, and R. Sagban, "Modified ACS Centroid Memory for Data Clustering," *J. Comput. Sci.*, vol. 15, no. 10, pp. 1439–1449, 2019, doi: 10.3844/jcssp.2019.1439.1449.

[28] J. Wahid and H. F. A. Al-Mazini, "Classification of Cervical Cancer Using Ant-Miner for Medical Expertise Knowledge Management," *Knowl. Manag. Int. Conf.*, no. November, 2018.

[29] 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.

[30] T. Stützle *et al.*, "Parameter adaptation in ant colony optimization," in *Autonomous Search*, vol. 9783642214, 2012, pp. 191–215.

[31] M. Maur, M. López-Ibáñez, and T. Stützle, "Pre-scheduled and adaptive parameter variation in MAX-MIN ant system," *2010 IEEE World Congr. Comput. Intell. WCCI 2010 - 2010 IEEE Congr. Evol. Comput. CEC 2010*, no. August, 2010, doi: 10.1109/CEC.2010.5586332.

[32] A. E. Eiben and S. K. Smit, "Parameter tuning for configuring and analyzing evolutionary algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1. pp. 19–31, 2011.

[33] 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.

[34] K. Velusamy and R. Manavalan, "Performance Analysis of Unsupervised Classification based on Optimization," *Int. J. Comput. Appl.*, vol. 42, no. 19, pp. 22–27, 2012, doi: 10.5120/5801-8090.

[35] K. M. Salama, A. M. Abdelbar, and A. a. Freitas, "Multiple pheromone types and other extensions to the Ant-Miner classification rule discovery algorithm," *Swarm Intell.*, vol. 5, no. 3–4, pp. 149–182, Jun. 2011, doi: 10.1007/s11721-011-0057-9.

[36] R. Forsati, A. Keikha, and M. Shamsfard, "An improved bee colony optimization algorithm with an application to document clustering," *Neurocomputing*, vol. 159, no. 1, pp. 9–26, 2015, doi: 10.1016/j.neucom.2015.02.048.

[37] B. Desgraupes, "Clustering Indices," *CRAN Packag.*, no. April, pp. 1–10, 2013, [Online]. Available: cran.r-project.org/web/packages/clusterCrit.

[38] J. Chavarria-Molina, J. J. Fallas-Monge, and J. Trejos-Zelaya, "Clustering via Ant Colonies: Parameter Analysis and Improvement of the Algorithm," 2019, [Online]. Available: http://arxiv.org/abs/1912.01105.

[39] S. Zhu and L. Xu, "Many-objective fuzzy centroids clustering algorithm for categorical data," *Expert Syst. Appl.*, vol. 96, pp. 230–248, 2018, doi: 10.1016/j.eswa.2017.12.013.

[40] M. H. Chehreghani, H. Abolhassani, and M. H. Chehreghani, "Improving density-based methods for hierarchical clustering of web pages," *Data Knowl. Eng.*, vol. 67, no. 1, pp. 30–50, 2008.

[41] M. Haghir, H. Abolhassani, and M. Haghir, "Improving density-based methods for hierarchical clustering of web pages," *Data Knowl. Eng.*, vol. 67, pp. 30–50, 2008.

[42] K. Bache and M. Lichman, "UCI Machine Learning Repository," *Univ. Calif. Irvine Sch. Inf.*, vol. 20, no. 8, 2013, doi: University of California, Irvine, School of Information and Computer Sciences.

[43] M. Kudo and J. Sklansky, "Comparison of algorithms that select features for pattern classifiers," *Pattern Recognit.*, vol. 33, no. 1, pp. 25–41, 2000, doi: 10.1016/S0031-3203(99)00041-2.

[44] T. Niknam and B. Amiri, "An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis," *Appl. Soft Comput. J.*, vol. 10, no. 1, pp. 183–197, 2010.