Received: March 8, 2019

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

U-control Chart Based Differential Evolution Clustering for Determining the Number of Cluster in *k* -Means

Ahmad Ilham[1]* Romi Satria Wahono[2] Catur Supriyanto[2] Adi Wijaya[3]

[1]*Department of Informatics, Universitas Muhammadiyah Semarang, Semarang, Indonesia*
[2]*Informatics Engineering Department, Universitas Dian Nuswantoro, Semarang, Indonesia*
[3]*Informatics Engineering Department, Universitas Mohammad Husni Thamrin, Jakarta, Indonesia*
* Corresponding author's Email: ahmadilham@unimus.ac.id

Abstract: Determining the cluster number in *k*-means is problematic since it affects the quality of cluster for numerous applications in the data mining. The automatic clustering differential evolution (ACDE) is one of the most used clustering methods that are able to determine the cluster number automatically. However, ACDE still makes use of the manual strategy to determine a value *k* activation threshold thereby affecting its performance. In this study, the u-control chart (UCC) method use to tackle the ACDE method problem. UCC method used for the initial step to get a value of the variables sought before initialization of the variable vector on ACDE. The UCC is a method from statistical process control (SPC) field which has proved to be effective in solving the problem of management control attributes. The performance of the proposed method was tested using seven public datasets from the UCI repository and Clustering basic benchmark repository and evaluated with Davies Bouldin Index (DBI) and CS measure. The results show that, the proposed method yields excellent performance compared to prior researches for most datasets with optimal cluster number yet lowest DBI and CS measure. It can be concluded that the UCC method is able to determine k activation threshold in ACDE that caused effective determination of the cluster number for *k*-means clustering.

Keywords: K-means, Automatic clustering, Differential evolution, *K* activation threshold, U-control chart.

1. Introduction

The *k*-means method is one of the hard partition methods in cluster analysis of the field of data mining. The *k*-means has advantages i.e. it is easy to implement, grouped a large dataset and stable performance across different problems [1, 2]. However, the clustering results of *k*-means depend on a certain number of clusters as inputs, if the estimated number of clusters does not tally with the final solution, the chances of clustering are very low [3, 4–7]. Meanwhile, getting the number of *k* as an input on *k*-means is still not an easy task because the user requires prior specification number of the cluster [8]. This condition is termed a local optimum problem [9]. In practice, the local optimum problem is overcome by applying the method several times with a different number of *k* then choosing the best results. Determining the number of clusters is significant for the *k*-means method [10]. Automatic clustering methods are one solution that helps the user determine the optimal number of clusters [11]. Therefore, the automatic clustering method is an effective solution to this problem.

Research on the determination of the number of clusters used automatic clustering methods which are based on Evolutionary Computation (EC) technique on *k*-means method has done a lot and has been published with different methods, namely Automatic Clustering Differential Evolution (ACDE) [12], combination methods between PSO and *k*-means on Dynamic Clustering with Particle Swarm Optimization (DCPSO) [13], Genetic Clustering for unknown *k* clustering (GCUK) [14], and harmonious genetic clustering algorithm (HGCA) [15]. The detailed comprehensive related

evolutionary computation for automatic number of clusters you can see [16].

Automatic clustering methods have been used to determine the number of clusters in the *k*-means but are yet to achieve an accurate cluster result. Therefore, it is necessary to improve the performance of automated grouping methods used for determining the number of clusters. The ACDE method is the most popular EC techniques which have effectively improved the performance of automatic clustering methods proposed by previous researchers [12]. ACDE predicated on differential evolution (DE) method is one of the strongest, fastest, and most efficient global search heuristics methods in the world that is very easy to use with high-dimensional data, it can be employed using polynomial functions and other functions because it is easy to change the values of control variables such as NP, F, and CR to obtain good search results [7]. However, ACDE has a weakness in determining *k* activation threshold that is still dependent on user judgment [17].

The ACDE was then developed by [18] with a combination of ACDE and *k*-means method, they call it ACDE-*k*-means. This method termed as the automatic clustering approach based on differential evolution method combining with *k*-means for crisp clustering method aimed at improving clustering performance. The ACDE method is capable of finding the number of clusters automatically and is able to balance the evolutionary process of DE methods to achieve better partitions than the classic DE. However, the DE classic method still depends on user's considerations to determine the *k* activation threshold thereby affecting the performance of the DE method [19].

The u-control chart (UCC) method is employed to determine the *k* activation threshold that is used for the initial step to get the value of the variables sought before initialization of the variable vector. The UUC is a method from statistical process control (SPC) which has proved to be effective in solving the problem of management control attributes [20], other methods such us p-control chart and c-control chart are methods but not used. This research focuses on UCC only. The UCC used to average the data to be measured is then reduced and added to find upper and lower bound values on the number of attributes to the searched variables. A product is said to have a good quality if the average value is at a threshold or the average value is between the upper and lower bound. Based on the above assumption, the data is good if it is within the threshold of the u-control chart.

The aim of this study is to apply the UCC method to determine *k* activation threshold on ACDE. Where ACDE is used to determine the number of clusters in *k*-means automatically and improve the performance of *k*-means.

This study is organized as follows. In section 2, there is an explanation of the related works. In section 3, there is a presentation of the proposed method. The experimental results of comparing the proposed method with others are given in section 4. Finally, the last section is devoted to concluding the work of this paper.

2. Literature review

Several studies have been carried out to find the number of clusters of *k*-means on automatic clustering evolutionary methods. The mostly used clustering methods, that is, the main methods, combined with evolutionary computation methods, are ACDE [12], ACDE-*k*-means [18], and HGCA [15]. So far, the clustering performance method to achieve optimal cluster number results is still a subject of further research because the best performance from all evaluations has not been completely achieved.

Determining the number of clusters has attracted much attention from the population-based optimization research community still a challenging problem that must be overcome. We first reviewed several strategies related to Differential Evolution to automatically determine *k*. In early 2018, Das *et al.* [12] tried to use DE to automatically determine the number of K in real-life datasets. The results of their experiments showed that the proposed method was superiority in all datasets compared with the DCPSO, GCUK, and Classical DE methods based on two DBI and CS measure evaluation indices. Following the research [12] flow, Kuo *et al.* [18] developed the Das e*t al.* [12] method by combining ACDE and *k*-Means, and they call it ACDE-*k*-means. They developed ACDE and *k*-means for crunchy grouping. In this case, ACDE uses the basic DE method which has weaknesses as explained in Chapter 1. The purpose of this method is to find the optimal number of clusters in *k*-means without knowing information from a priori data. The two evaluation indices used were CS measure and VI index, then, the dataset tested came from the UCI: Iris and Wine repository, while the comparison method used was DE Classic showing superior ACDE-*k*-means from DE Classic in all datasets.

In addition to DE-based techniques, other population-based optimization techniques, such as GAS are also often used to automatically determine

k. Recently, Huang *et al*. [15] has proposed a genetic approach based on harmonious marriage in eugenic theory to produce more quality clusters while grouping data samples, called HGCA. HGCA aims to choose the most suitable partner for each chromosome and consider chromosomes, gender, age, and fitness when calculating mating attraction.

3. Materials and the proposed method

3.1 Clustering problem definition

Clustering problem can be defined as a data set $M = \{m_1, m_2, \dots, m_n\}$, which contains n data points in d dimension and the dataset will be grouped into *k* number of clusters $C = \{c_1, c_2, \dots, c_k\}$. There are three properties that should be maintained by hard partition problems:

1. each cluster should possess at least one data point assigned, i.e., Eq. (1).
2. no data point is common to two different clusters, i.e., Eq. (2).
3. each pattern must be attached to a cluster as indicated, i.e., Eq. (3).

$$C_i \neq \emptyset, \quad i = 1,2, \dots, K \tag{1}$$

$$C_i \cap C_j = \emptyset, \quad i,j = 1,2, \dots, K \; and \; i \neq j \tag{2}$$

$$\cup_{i=1}^{K} C_i = M \tag{3}$$

The above three properties boil down to one question, that is, how to determine the optimal cluster number $C = \{C^1, C^2, \dots, C^{M(m,k)}\}$, where Eq. (4) is the number of feasible partitions and Eq. (5) is the same as optimize.

$$M(m,k) = \frac{1}{k_i}\sum_{i=1}^{k}(-1)^i \binom{k}{i}(k-1)^i \tag{4}$$

$$Optimize \; \underset{\boldsymbol{C}}{f(M_{n\times d}, C)} \tag{5}$$

Where, f is a mathematics-statistical function that determines the goodness of a partition premised on the distance measure of the patterns, while C is a single partition from the set $\boldsymbol{C}$.

3.2 Cluster validity indexes

The aim of the cluster validity index is to measure how efficient cohesion and separation are [11]. Compactness is used to measure variation or pattern of data within a cluster and separation shows cluster isolation from each other using matrix distance (usually their used Euclidean distance). There are many indices the validity index cluster can use, but in this study, only DBI /CS measure is used as clusters validity index to help find the right number of clusters because it has been widely used and is state-of-the-art. In this study, the same cluster validity (DBI /CS measure) original ACDE method is used as fitness function [17].

3.2.1. Davies Bouldin index (DBI)

The DBI was introduced by Davies and Bouldin [21]. It aims to determine how well clustering has been done by evaluating the quantity and attribute derived from the dataset. This index calculates the ratio between cluster cohesion and cluster separation. The formulas are as follows Eq. (6)-(9).

1. First, define the within i^{th} cluster cohesion

$$S_{i,g} = \left[\frac{1}{B_i}\sum_{\vec{s}\in C_i}\|\vec{x} - \vec{m}_i\|_2^g\right]^{1/g} \tag{6}$$

$$Z_{i,j,t} = \left\{\sum_{p=1}^{z}\left|m_{i,p} - m_{j,p}\right|^t\right\}^{\frac{1}{t}} = \left\|m_i, m_j\right\|_t \tag{7}$$

Here $\vec{m}_i$ is the i^{th} cluster center, $g, t \geq 1$, g is an integer, g and t can be independently selected. B_i is the number of elements in the i^{th} clusters C_i.

2. Secondly, define between i^{th} and j^{th} cluster distance, see Eq. (7)
Where, $Z_{i,j,t}$ is a separation between cluster C_i and C_j. $m_{i,p}$ is the p^{th} element of $\vec{m}_i$ and there is Z such elements in m for it is a Z dimensional centroid. Next, $R_{i,gt}$ is defined as:

$$R_{i,gt} = \underset{j \in K, j \neq i}{max}\left\{\frac{S_{i,g}+S_{j,g}}{Z_{ij,t}}\right\} \tag{8}$$

$R_{i,g,t}$ is a measure of how good the clustering schema is $Z_{ij,t}$. $Z_{ij,t}$ is a separation between the i^{th} and j^{th}.

3. Finally, DBI measure is defined

$$DBI(K) = \frac{1}{K}\sum_{i=1}^{K} R_{i,gt} \tag{9}$$

the lowest value of $DBI\ (K)$ is a valid optimal clustering.

3.2.2. CS measure

The CS measure proposed by [22] is a simple clustering measurement index which can give more cluster centroids to the area that has lower density data than conventional clustering methods [18]. First, the cluster centroid of a cluster is found using

the average data vector which belongs to that cluster as shown in Eq. (10). Afterward, CS measure is calculated using Eq. (11), where the distance metrics between two data point $\vec{X}_i, \vec{X}_j$ are represented by $d(\vec{X}_i, \vec{X}_j)$.

$$\vec{m}_i = \frac{1}{N_i}\sum_{x_j \in C_i} \vec{x}_j \tag{10}$$

$$CS(K) = \frac{\frac{1}{K}\sum_{i=1}^{K}\left[\frac{1}{N_i}\sum_{\vec{X}_i \in C_i} \max_{\vec{X}_q \in C_i}\{d(\vec{X}_i, \vec{X}_q)\}\right]}{\frac{1}{K}\sum_{i=1}^{K}\left[\max_{j \in K, j \neq i}\{d(\vec{m}_i, \vec{m}_j)\}\right]} = \frac{\sum_{i=1}^{K}\left[\frac{1}{N_i}\sum_{\vec{X}_i \in C_i} \max_{\vec{X}_q \in C_i}\{d(\vec{X}_i, \vec{X}_q)\}\right]}{\sum_{i=1}^{K}\left[\max_{j \in K, j \neq i}\{d(\vec{m}_i, \vec{m}_q)\}\right]} \tag{11}$$

3.3 The *k*-means method

In brief, the *k*-means method attempts to find non-overlapping clusters which are represented by centroids. The *k*-means steps are as follows:

1. Assign the *k* center cluster to user randomly to be the starting cluster center,
2. Select a cluster (*k*) center randomly to be the starting cluster center,
3. Allocate all data to the nearest cluster center with distance matrix (Euclidean distance),
4. Recalculate the new cluster center based on data following each cluster,
5. Repeat steps 3 and 4 until a convergent condition is met or no data moves from one cluster to another

3.4 Differential evolution method

Differential evolution (DE) method was proposed by [23]. DE is a stochastic search method and a population-based search predicated on generating population dots to achieve a minimum of a function. The basic idea of DE is first, application of mutation to produce experimental vectors (trials vector), then trial vectors are employed in the crossover process to produce offspring and step size in mutations not sampled or indexed from known population distributions. There are four steps in DE [7], namely initialization, mutation, crossover and selection.

Initialization. Before initialization of the vector is searched, it is important to determine *k* activation threshold lower ($lb_{min,j}$) and upper bound ($ub_{min,j}$). *K* activation threshold will be utilized as the initial step of generating the value of the variable searched [0, 1]. Then, do the initialization process $P_{i,d}(t)$ (randomly generated) which the initial population is based on [18]. The i^{th} is individual vector chromosome i^{th} of the population at time generation step t has d components (dimensions) as shown in Eq. (12). Furthermore, the initial value generation of first variable j^{th} and i^{th} vector is shown in Eq. (13). Then, it will flourish using mutation and crossover.

$$P_{i,d}(t) = P_{i,1}(t), P_{i,2}(t), \dots, P_{i,d}(t) \tag{12}$$

$$P_{i,j}(0) = lb_{min,j} + rand_{i,j}(0,1).(ub_{max,j} - lb_{min,j}) \tag{13}$$

Mutation. After initialization, DE will cause the mutation and combine the initial population to have a population with the size of the *N* trial vector. The Trial vector is defined as $Z_{i,d}(t+1)$. In DE, mutations are done by adding two vectors $P_{j,d}(t)$, $P_{k,d}(t)$ differences to the third vector $P_{l,d}(t)$ with Eq. (14). Differences of two vectors are selected by random need to be scaled first before being added to the third vector $P_{l,d}(t)$ to put population growth rates under control.

$$Z_{i,d}(t+1) = P_{j,d}(t) + F(P_{k,d}(t) - P_{l,d}(t)) \tag{14}$$

Crossover. At this stage, DE crossed each vector $V_{i,d}(t)$ with a mutant vector $Z_{i,d}(t+1)$ to produce the vector of the crosses with Eq. (15).

$$U_i(t+1) = U_{ji,d}(t+1) = \begin{cases} Z_{i,d}(t+1)\ if\ rand_j(0,1) \leq CR\ or\ j = rand(d) \\ V_{i,d}(t)\ if\ rand_j(0,1) > CR\ or\ j \neq rand(d) \end{cases} \tag{15}$$

Selection. Finally, to get a new offspring $V_i(t+1)$, the trial vector $U_i(t+1)$ will have compared to the objective function $f(V_i(t))$. If the trial vector $U_i(t+1)$ has a goal function value that is not as big as its objective function vector target $f(V_i(t)$ after that $U_i(t+1)$ will replace the position $f(V_i(t)$ in the population in the next generation. If the opposite happens, the target vector $V_{i,d}(t)$ will remain in its position in the population. Then Mutation, crossover and selection operations will continue until some stopping criteria are reached.

$$V_i(t+1) = \begin{cases} U_i(t+1) & if\ f(U_i(t+1) > f(V_i(t))) \\ V_{i,d}(t) & if\ f(U_i(t+1) \leq f(V_i(t))) \end{cases} \tag{16}$$

3.5 Objective function

Objective function is a simulated search on a data set to guide towards an optimal global solution. In the case of clustering problems, the objective function usually uses the cluster validity index [11]. In this case, DBI and CS measure are used as objective function based on the finding of [12] as follows Eq. (17) and Eq. (18).

$$f1 = \frac{1}{CS_i(K)+eps}, \tag{17}$$

The eps is a small bias term equal to 2×10^{-6} near zero. 2×10^{-6} is a cluster k for k with set number of clusters as initialization to cluster of the datasets.

$$f2 = \frac{1}{DBI_i(K)+eps}, \tag{18}$$

Where DBI_i is the DB index, evaluated on the partitions yielded by the i^{th} vector and eps is the same as before.

3.6 Proposed *k* activation threshold of differential evolution method

The differential evolution (DE) method has a weakness in sample size population or *k* activation threshold which is used to determine what is often specified by the user. However, in this condition, it is found that the inappropriate choice of the population size may hamper the performance of DE method [19]. To solve this problem, u-control chart (UCC) is used as a proposed method to improve DE. The u-control chart is a type of control chart in statistical quality control, which is used to regulate the process and ensure quality [24]. There are three process steps for *k* activation threshold on DE, and they can be defined as follows:

$$\bar{u} = \sum_{x_i} h \tag{19}$$

$$ub = \bar{u} + K\sqrt{\frac{\bar{u}}{n_i}} \tag{20}$$

$$lb = \bar{u} - K\sqrt{\frac{\bar{u}}{n_i}} \tag{21}$$

3.7 The proposed method

In this research, a combination of the u-control chart (UCC) and differential evolution clustering automatic method is proposed to determine the number of clusters on *k*-means (UCC+ACDE-*k*-means). The aim of the UCC method is to control *k* activation threshold of the differential evolution clustering automatic method. The latter will search the optimal number of clusters in the data automatically as required by *k*-means. The representation of chromosome used is based on [14]. Because the differential evolution clustering automatic method produces a premature cluster, the *k*-means is implemented to repair the premature clustering.

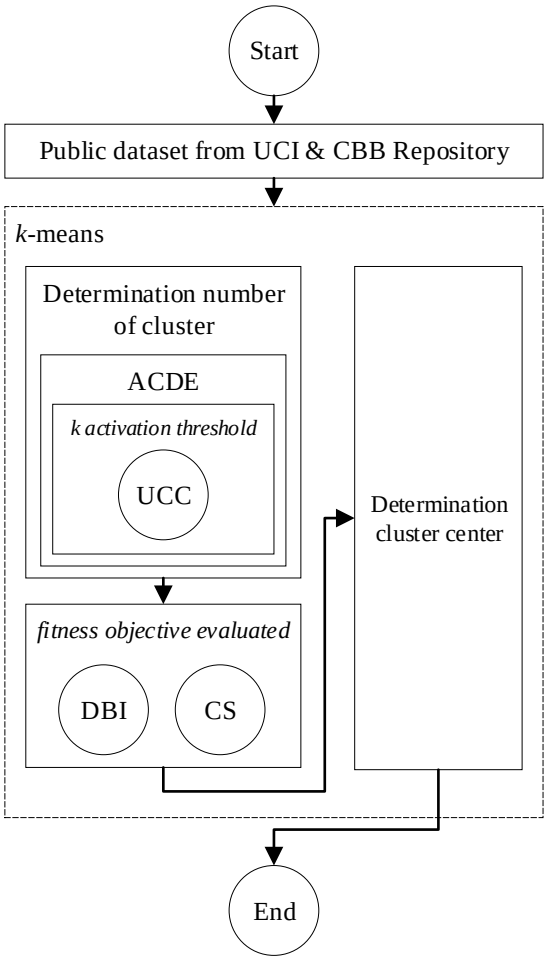

Figure.1 Blog diagram of the proposed method

As shown in Fig. 1 the steps for the complete proposed method are given here.

1. Prepare datasets.
2. Initialize each chromosome containing a selected number of *k* randomly selected clusters and specify the *k* activation threshold using the UCC method defined with a stage as follows Eq. (19), (20) and (21). In Eq. (19), the average value is given by the average of all attributes. After that Eq. (20), the upper bound (ub) is calculated. Next Eq. (21), the lower bound (lb) is calculated.

3. Generate initial population randomly based on predetermined *k* activation threshold values.
4. Find the active cluster center, which is defined as shown in Eq. (22).

$$IF_{v_{i,k}} > 0{,}5\ THEN\ cluster\ center\ v_{i,k}m_k\ is\ ACTIVE\ ELSE\ v_{i,k}m_k\ is\ INACTIVE \quad (22)$$

Where the center of the $v_{i,k}$ cluster on the chromosome will be active or selected if $v_{i,k}T_k > 0{,}5$. Conversely, if $v_{i,k}T_k < 0{,}5$ the center of the cluster $v_{i,k}$ is not active in the i-th chromosome. The $v_{i,k}T_k$ is the cost population of the data generation, while the best solution cost or $v_{i,k}m_k$ is the best solution for each iteration.

5. For iteration
   a. Find the distance of each data vector from all active centroids of the i^{th} chromosome,
   b. Allocate each data vector to a cluster with the shortest distance,
   c. Change member(s) of the population (based on DE method) using the objective function to make the selected population better,
   d. Apply *k*-means method. The active cluster number is used as input *k*-means to adjust i^{th} active chromosome.
6. As a result, the minimum objective is the output of the global best chromosome.

4. Results and discussion

The experiments were conducted using a computing platform with Intel Celeron 2.16 GHz CPU, 8 GB RAM and Microsoft Windows 10 Home 64-bit used as the operating system and MATLAB version R2016a used as the data analytics tool. MATLAB would produce a model performance as the calculation output, such as average value best cluster DBI and CS measure. The proposed method was tested using artificial dataset [15, 25] include S1 (300, 2, 6), S2 (500, 2, 9), S3 (5000, 2, 15), S4 (5000, 2, 15) and real world dataset that is Iris (150, 4, 3), Vowel (871, 36, 6), and Letter (2000, 16, 26) from UCI Machine Learning Repository [15, 26].

Parameter setting for proposed method based on the recommendation by [12] is as follows: $max - iter = 200$, $pop - size = 10 * dim$, $CRmax = 1.0$ and $CRmin = 0.5$. Max-iteration indicates the amount of iteration, pop-size is the size of the population, crossover probability is used to initialize the position of a particle or chromosome.

4.1 The ACDE-*k*-means without UCC method

First, an experiment was conducted of all datasets using only ACDE k-means without the UCC method. Classes on the data were omitted to analyze optimal partition in a data. The experimental results are shown in Table 1. This method produced three DBI mean value with an excellent mean and CS measure mean value also getting three excellent same means. Meanwhile, DBI varied from 0.039-1.273 and CS measure varied from 0.042-1.168. In the case of this method based on the number of *k* cluster exactness, DBI mostly good in 4 from 7 datasets and CS measure is also good in 6 from 7 datasets. Based on this result, the method is promising enough since it still produced between 4 and 6 excellent results for all datasets.

4.2 The proposed method with u-control chart method

In the second experiment, the u- control chart (UCC) method was implemented to resolve the problem of *k* activation threshold automatically without requiring the user to enter the required values in DE for ACDE-*k*-means in determining the number of clusters of *k*-means, whereas, *k*-means was implemented to do repair grouping. The experimental results are reported in Table 2.

As you can see in Table 2, the method produced 9 DBI and 11 CS measure with an excellent value. Meanwhile, DBI results vary from 0.309-1.223 and CS measure results vary from 0.304 - 1.193. In the case of DBI and CS measure, the method mostly produced a fair average cluster and got correctly the number of clusters of all datasets. Based on this result, the method is still promising enough since it still produced 9 DBI and 11 CS measure with excellent value.

A more detailed comparison of the first and the second experiment is presented in Table 3. The best model automatic clustering on each dataset is highlighted with boldfaced print and the best optimal cluster result is marked with (*) as optimal and (#) not optimal of (1) and (2) squared on each dataset. As shown in Table 3, the second experiment UCC+ACDE-*k*-means based DBI objective function evaluation outperforming only 5 from 7 datasets, and the optimal search *k* results of both methods are extremely good except for the S3, and Iris dataset. The same thing happened in the CS measure is objective function evaluation is the proposed method is superior in almost all datasets except Iris and Letter, and the optimal search *k*

Table 1. Summary of performance measurement based on objective function using DBI and CS Measure for all datasets of ACDE-k-means only.

Datasets	Class Optimal (*k*)	Mean Validity Index and Cluster Number Optimal			
		DBI	k	CS	k
S1	6	0.583	6*	0.891	6*
S2	9	0.537	9*	0.598	9*
S3	15	0.591	18#	0.809	15*
S4	15	0.703	15*	0.645	15*
Iris	3	0.039	5#	0.042	2#
Vowel	6	1.273	6*	0.982	6*
Letter	26	1.029	22#	1.168	28*

*number cluster optimal #not optimal

Table 2. Summary of performance measurement based on objective function using DBI and CS Measure for all datasets of proposed method.

Datasets	Class Optimal (*k*)	Mean Validity Index and Cluster Number Optimal			
		DBI	k	CS	k
S1	6	0.511	6*	0.609	6*
S2	9	0.495	9*	0.512	9*
S3	15	0.487	13#	0.786	15*
S4	15	0.692	15*	0.581	15*
Iris	3	0.309	4#	0.304	2#
Vowel	6	1.032	6*	0.821	6*
Letter	26	1.223	26*	1.193	26*

*number cluster optimal #not optimal

Table 3. Results comparison ACDE-k-means only vs proposed method

Datasets	Class optimal (*k*)	Mean validity index and number cluster optimal							
		DBI		*k*		CS Measure		*k*	
		(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)
S1	6	0.583	**0.511**	6*	6*	0.891	**0.609**	6*	6*
S2	9	0.537	**0.495**	6#	9*	0.598	**0.512**	6*	9*
S3	15	0.591	**0.487**	18#	13#	0.809	**0.786**	15*	15*
S4	15	0.703	**0.692**	15*	15*	0.645	**0.581**	15*	15*
Iris	3	**0.039**	0.309	5#	4#	**0.042**	0.304	2#	2#
Vowel	6	1.273	**1.032**	6*	6*	0.982	**0.821**	6*	6*
Letter	26	**1.029**	1.223	22#	26*	**1.168**	1.193	28*	26*

(1)ACDE-*k*-means only (2)Proposed method ; *Number cluster optimal #not optimal

Table 4. Comparison to prior search based on DBI and CS as objective function for all datasets

Dataset	Objective function	Methods			
		ACDE	ACDE-k-means	HGCA	Proposed method
S1	DBI	0.894	0.583	0.532	**0.511**
	CS Measure	1.525	0.891	**0.363**	0.609
S2	DBI	0.910	0.537	0.580	**0.495**
	CS Measure	0.731	0.598	0.594	**0.512**
S3	DBI	0.716	0.591	0.544	**0.487**
	CS Measure	0.973	0.809	0.886	**0.786**
S4	DBI	0.831	0.703	**0.644**	0.692
	CS Measure	0.774	0.645	0.695	**0.581**
Iris	DBI	0.465	**0.039**	0.414	0.309
	CS Measure	0.664	**0.042**	0.613	0.304
Vowel	DBI	0.992	1.273	**0.632**	1.032
	CS Measure	0.909	0.982	1.021	**0.821**
Letter	DBI	1.718	**1.029**	1.235	1.223
	CS Measure	1.278	1.168	1.322	**1.193**

results of both methods are extremely good only except for Iris dataset.

Based on this study, overall the second experiment outperformed and is better than the first experiment where DBI and CS measure used as objective function for finding the number of the optimal cluster.

4.3 Proposed method with others prior research

Finally, the proposed method was compared with prior research such as ACDE [12], ACDE-*k*-means [16], and HGCU [13]. Table 3 shows the results of the experimental summary has been composed based on DBI and CS measure as the objective function of each method for each dataset. The purpose of summarizing the methods is to identify the best method by looking at the average value of an objective function of DBI and CS measure. A method with the lowest value of DBI and CS measure is the best method. Table 4 shows the comparison of prior research with the proposed method in all datasets.

As indicated in Table 4, all existing methods have their complexity using evolutionary strategy automatic clustering methods for determining the number of clusters automatically in *k*-means, while the proposed method uses technical-statistical control for problem-solving of ACDE as an automatic clustering strategy for finding the optimal *k* clusters in *k*-means method. Contrast with Table 5, which that, we only use two datasets as comparing testing results.

After knowing the performance of each method for each dataset, the next step is to differentiate tests between methods using a non-parametric statistical test. Non-parametric statistical tests have been widely recommended for use in the field of evolutionary research in clustering [7]. On the premise of this recommendation, in the framework of this study, Friedman's test is used to compare the DBI and CS measure values of all methods. Friedman's test is predicated on the performance of the mean of rank (R) clustering method on each dataset.

4.4 Differentiating test using statistic calculation between the proposed and prior research

Friedman test is used to significantly differentiate tests between the proposed method with *k*-means classic, GCUK, DCPSO, ACDE, and ACDE-*k*-means. In the Friedman test as a statistical significant testing, the p-value is used to obtain the statistical test that is actually observed, with an assumption that the null hypothesis is true. The null hypothesis is often rejected when the p-value is less than the predetermined significance level (α). In this case, the statistical significance level (α) is set at 0.05. This means that there is a statistically significant difference if p-value < 0.05 so that way, one may proceed with a Nemenyi post-hoc test to detect which particular agglomeration differ considerably. When p-value > 0.05 it means that there is no statistically significant difference.

Friedman test will indicate the ranking of method performance for each dataset [27], where rank (R) 1 shows the best method, rank 2 shows the second-best method and so on. In this study, Friedman statistical test is employed to compare two or more clustering methods over multiple datasets. Here, it is employed to compare the DBIs and the different automatic clustering methods. This test is based on the average ranked (R) performances of the classification methods on each dataset.

Let r_i^j be the rank of the j^{th} of C method on the j of D dataset. The Friedman test compares the average rank of the method $R_j = \frac{1}{D}\sum_{i-1}^{D} R_i^j$. Under the null hypothesis, all the methods are equivalent and so their ranks R_j should be equal. The Friedman statistic is calculated as follows and distributed according to X_F^2 with $C-1$ degrees of freedom when D and C are big enough.

$$X_F^2 = \frac{12D}{C(C+1)}\left[\sum_j^D R_j^2 - \frac{C(C+1)}{4}\right] \quad (23)$$

If the null-hypothesis is rejected, the next thing is to proceed with a post-hoc test. The Nemenyi test is used to compare all classifiers with each other. The performance of the two classifiers is considerably different if the corresponding average ranks differ by at least the critical difference, given by:

$$CD = q_a\sqrt{\frac{C(C+1)}{D}} \quad (24)$$

where, critical values are based on the Student size range statistic.

4.4.1. Davies Bouldin index comparison with prior research

Friedman test was employed to significantly differentiate tests between the proposed method with ACDE, ACDE-*k*-means, and HGCA. For the testing experiment of Friedman test results based on DBI, obtained p-value of 0.025. This value is lesser than the level of significance α = 0.05, so it can be concluded that there is a significant difference between the methods. Table 5 shows the DBIs of the proposed method and prior research. The last record of Table 5 indicates the mean rank (***R***) of each method over all datasets based on the Friedman test.

The best methods in each dataset are highlighted with boldfaced print and underline. As shown in Table 6, the proposed method has the lowest

Table 5. Comparison of results between the proposed method and prior research based on DBI, CS Measure & Friedman rank test on all datasets

Dataset	ACDE		ACDE-k-means		HGCA		Proposed	
	DBI	***CS***	***DBI***	***CS***	***DBI***	***CS***	***DBI***	***CS***
S1	0.894	1.525	0.583	0.891	0.532	**0.363**	**0.511**	0.609
S2	0.910	0.731	0.537	0.598	0.580	0.594	**0.495**	**0.512**
S3	0.716	0.973	0.591	0.809	0.544	0.886	**0.487**	**0.786**
S4	0.831	0.774	0.703	0.645	0.644	0.695	**0.692**	**0.581**
Iris	0.465	0.664	**0.039**	**0.042**	0.414	0.613	0.309	0.304
Vowel	0.992	0.909	1.273	0.982	**0.632**	1.021	1.032	**0.821**
Latter	1.718	1.278	**1.029**	**1.168**	1.235	1.322	1.223	1.193
M	0.932	0.979	0.679	0.734	**0.654**	0.785	0.678	**0.687**
R	3.714	3.571	2.429	2.143	2.143	2.857	**1.714**	**1.429**

Table. 6 Pairwise comparison nemenyi post hoc test

	ACDE	**ACDE-k-means**	**HGCA**	**Proposed**
ACDE	0	1.429	0.714	2.143
ACDE-k-means	-1.429	0	-0.714	0.714
HGCA	-0.714	0.714	0	1.429
Proposed	-2.143	-0.714	-1.429	0
Critical different: 1.7728				

Table. 7 P-value of Nemenyi post hoc test

	ACDE	**ACDE-k-means**	**HGCA**	**Proposed**
ACDE	1	0.163	0.729	**0.010**
ACDE-k-means	0.163	1	0.729	0.729
HGCA	0.729	0.729	1	0.163
Proposed	**0.010**	0.729	0.163	1

Friedman score (***R***). Below the (R) row of Table 6, DBI's mean (***M***) is shown.

4.4.1. CS measure comparison with prior research

For the testing experiment of the Friedman test results based on CS measure, obtained a p-value of 0.013. This value is smaller than the level of significance $\alpha = 0.05$, so it can be concluded that there is a significant difference between the methods. Table 5 reports the CS measure of the proposed method and prior research. The last record of Table 5 indicates the mean rank (***R***) of each method over all datasets based on the Friedman test.

The best automatic clustering method on each dataset is highlighted with boldface print. Table 5 shows that the proposed method has the lowest CS measure value of $\boldsymbol{M} = 0.687$ of all the datasets for each method. For the Friedman test results of the mean of rank (***R***), the proposed method has the best rank $\boldsymbol{R} = 1.429$ superior to another comparison method. Because there is a considerable difference between the proposed method and other comparative methods, the analysis will be continued by using a pairwise comparison Nemenyi post-hoc test. A comparison of Nemenyi post-hoc test is carried out to identify significantly different methods in which this test calculates all pairwise comparisons between clustering methods, if the value of pairwise comparison results is greater than the Critical Difference (CD) value, then there is a significant difference between others methods. Table 6 shows the ACDE-*k*-means, HGCA and the proposed method obtained values of 1.429, 0.714, and 2.143 which is bigger than CD = 1.7728, therefore it can be concluded that there is a significant difference between several methods. To calculate CD see Eq. (23) and Eq. (24). Furthermore, to find out which methods are significantly different, then continue to go through the p-value results in the Nemenyi post-hoc test. If the p-value < 0.05 then the performance of the method differs significantly as shown in Table 7. The p-value results of Nemenyi post-hoc test is shown in Table 7.

As shown in Table 7 p-value < 0.05 results are highlighted with boldfaced print, which means that

there is a statistically significant difference between ACDE method and the proposed method. Based on this result, it can be concluded that a combination of u-control chart (UCC) method and automatic clustering differential evolution (ACDE) method improve the performance of *k*-means by getting p-value (0.013) < 0.05. Furthermore, the proposed method is also superior to other methods such as HGCA. However, the proposed method cannot be said to have a superior performance over the ACDE because the performance of this method can be enhanced by adjusting the appropriate parameters and improve activation schema, and this has been confirmed by [17] that the performance of the ACDE methods still can be improved by setting the appropriate parameters right. The explanation for this observation is clear that for datasets with clusters that are easily seen with fair optimization capabilities can efficiently find cluster structures with DBI and CS measures provided. From the above observations, we conclude that for datasets with clear cluster structures or ambiguous cluster structures, the proposed method performs better than the other three methods in terms of getting the right number of clusters *k*.

5. Conclusion

In this study, we have presented a novel method our proposed called UCC-ACDE-*k*-means to automatically select the number of clusters *k*, where the clusters resulted are better than those produced by the state-of-the-art methods. Our main finding is that traditional Differential Evolution's (DE) method on the ACDE method for determining the number of automatic clusters in *k*-means still manually determine *k* activation thresholds so that will likely fail by falling into an undesired stagnation condition. Stagnation is an undesirable effect that occurs when population-based algorithms do not blend into (even suboptimal) solutions while population diversity is still high. Motivated by statistical quality control which is often used to control sample data for control diagrams in a multi-stage process. A product is said to have good quality if the average value is at the threshold or the average value is at the upper and lower limits. With this assumption, data is categorized as good if it is still within the U-control chart (UCC) threshold. Our method uses a hybrid method between UCC and ACDE to find a number of clusters of *k*-means, thus, the cluster results are no longer trapped into the local minimum and can increase clustering performance with the lowest cluster validity value. We have conducted extensive experiments to evaluate our proposed method on real-life and artificial datasets. Finally, the results show that our proposed method can find a number of clusters automatically without knowing the number of clusters in advance. We have also compared our proposed with some existing state-of-the-art methods. The results confirm that our approach is more effective and efficient for data clustering.

Further research may be added to other control chart methods from statistical process control (SPC) such as p-control chart (PCC) and c-control chart (CCC). According to [20], the SPC method can easily detect changes in the data in a process which may affect the quality of the results.

References

[1] S. B. Salem, S. Naouali, and Z. Chtourou, “A fast and effective partitional clustering algorithm for large categorical datasets using a k -means based approach”, *Comput. Electr. Eng.*, Vol. 68, No. 4, pp. 463–483, 2018.

[2] S. Chakraborty and S. Das, “Simultaneous variable weighting and determining the number of clusters—A weighted Gaussian means algorithm”, *Stat. Probab. Lett.*, Vol. 137, No. 6, pp. 148–156, 2018.

[3] M. A. Rahman and M. Z. Islam, “A hybrid clustering technique combining a novel genetic algorithm with K-Means”, *Knowledge-Based Syst.*, Vol. 71, No. 17, pp. 345–365, 2014.

[4] M. A. Rahman, M. Z. Islam, and T. Bossomaier, “ModEx and Seed-Detective: Two novel techniques for high quality clustering by using good initial seeds in K-Means”, *J. King Saud Univ. - Comput. Inf. Sci.*, Vol. 27, No. 2, pp. 113–128, 2015.

[5] A. Ilham, D. Ibrahim, L. Assaffat, and A. Solichan, “Tackling Initial Centroid of K-Means with Distance Part (DP-KMeans)”, In: *Proc. of 2018 International Symposium on Advanced Intelligent Informatics*, Vol. 1, pp. 185–189, 2018.

[6] M. A. Masud, J. Z. Huang, C. Wei, J. Wang, I. Khan, and M. Zhong, “I-nice: A new approach for identifying the number of clusters and initial cluster centres”, *Inf. Sci. (Ny).*, Vol. 466, pp. 129–151, 2018.

[7] M. Ramadas, A. Abraham, and S. Kumar, “FSDE-Forced Strategy Differential Evolution used for data clustering”, *J. King Saud Univ. - Comput. Inf. Sci.*, Vol. 31, No. 1, pp. 52–61, 2019.

[8] Y. Zhang, J. Mańdziuk, C. H. Quek, and B. W. Goh, “Curvature-based method for determining

the number of clusters", *Inf. Sci. (Ny).*, Vol. 415–416, pp. 414–428, 2017.

[9] C. Tîrnăucă, D. Gómez-Pérez, J. L. Balcázar, and J. L. Montaña, "Global optimality in k - means clustering", *Inf. Sci. (Ny).*, Vol. 439–440, pp. 79–94, 2018.

[10] W. Xiang, N. Zhu, S. Ma, X. Meng, and M. An, "A dynamic shuffled differential evolution algorithm for data clustering", *Neurocomputing*, Vol. 158, pp. 144–154, 2015.

[11] A. José-García and W. Gómez-Flores, "Automatic clustering using nature-inspired metaheuristics: A survey", *Appl. Soft Comput.*, Vol. 41, pp. 192–213, 2016.

[12] S. Das, A. Abraham, and A. Konar, "Automatic Clustering Using an Improved Differential Evolution Algorithm", *IEEE Trans. Syst. Man, Cybern. - Part A Syst. Humans*, Vol. 38, No. 1, pp. 218–237, 2008.

[13] M. G. H. Omran, A. Salman, and A. P. Engelbrecht, "Dynamic clustering using particle swarm optimization with application in image segmentation", *Pattern Anal. Appl.*, Vol. 8, No. 4, pp. 332–344, 2006.

[14] S. Bandyopadhyay and U. Maulik, "Genetic clustering for automatic evolution of clusters and application to image classification", *Pattern Recognit.*, Vol. 35, No. 6, pp. 1197–1208, 2002.

[15] F. Huang, X. Li, S. Zhang, and J. Zhang, "Harmonious genetic clustering", *IEEE Trans. Cybern.*, Vol. 48, No. 1, pp. 199–214, 2018.

[16] E. Hancer and D. Karaboga, "A comprehensive survey of traditional, merge-split and evolutionary approaches proposed for determination of cluster number", *Swarm Evol. Comput.*, Vol. 32, pp. 49–67, 2016.

[17] H.-H. Tam, S.-C. Ng, A. K. Lui, and M.-F. Leung, "Improved activation schema on Automatic Clustering using Differential Evolution algorithm", In: *Proc. of 2017 IEEE Congress on Evolutionary Computation*, pp. 1749–1756, 2017.

[18] R. Kuo, S. Erma, and A. Yasid, "Automatic Clustering Combining Differential Evolution Algorithm and k-Means Algorithm", In: *Proc. of the Institute of Industrial Engineers Asian Conference 2013*, pp. 1207–1215, 2013.

[19] A. P. Piotrowski, "Review of Differential Evolution population size", *Swarm Evol. Comput.*, Vol. 32, pp. 1–24, 2017.

[20] I. Kaya, "A genetic algorithm approach to determine the sample size for attribute control charts", *Inf. Sci. (Ny).*, Vol. 179, No. 10, pp. 1552–1566, 2009.

[21] D. L. Davies and D. W. Bouldin, "A Cluster Separation Measure", *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. PAMI-1, No. 2, pp. 224–227, 1979.

[22] C.-H. Chou, M.-C. Su, and E. Lai, "A new cluster validity measure and its application to image compression", *Pattern Anal. Appl.*, Vol. 7, No. 2, pp. 205–220, 2004.

[23] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", *J. Glob. Optim.*, Vol. 11, No. 4, pp. 341–359, 1997.

[24] A. Costa and S. Fichera, "Economic statistical design of ARMA control chart through a Modified Fitness-based Self-Adaptive Differential Evolution", *Comput. Ind. Eng.*, Vol. 105, pp. 174–189, 2017.

[25] P. Fränti and S. Sieranoja, "K-means properties on six clustering benchmark datasets", *Appl. Intell.*, Vol. 48, No. 12, pp. 4743–4759, 2018.

[26] D. Aha *et al.*, "UCI Repository of Machine Learning Database", 1987.

[27] J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets", *J. Mach. Learn. Res.*, Vol. 7, pp. 1–30, 2006.