Modern Applied Science; Vol. 12, No. 2; 2018
ISSN 1913-1844 E-ISSN 1913-1852
Published by Canadian Center of Science and Education

An Improved Version of K-medoid Algorithm using CRO

Amjad Hudaib[1], Mohammad Khanafseh[1] & Ola Surakhi[1]

[1] University of Jordan, King Abdullah II School for Information Technology, Computer Science Department, Amman, Jordan

Correspondence: Ola Surakhi, University of Jordan, King Abdullah II School for Information Technology, Computer Science Department, Amman, Jordan. E-mail: ahudaib@ju.edu.jo, mkhanafsa@gmail.com, ola.surakhi@gmail.com

Received: January 8, 2017 Accepted: January 18, 2018 Online Published: January 30, 2018

doi:10.5539/mas.v12n2p116 URL: https://doi.org/10.5539/mas.v12n2p116

Abstract

Clustering is the process of grouping a set of patterns into different disjoint clusters where each cluster contains the alike patterns. Many algorithms had been proposed before for clustering. K-medoid is a variant of k-mean that use an actual point in the cluster to represent it instead of the mean in the k-mean algorithm to get the outliers and reduce noise in the cluster. In order to enhance performance of k-medoid algorithm and get more accurate clusters, a hybrid algorithm is proposed which use CRO algorithm along with k-medoid. In this method, CRO is used to expand searching for the optimal medoid and enhance clustering by getting more precise results.

The performance of the new algorithm is evaluated by comparing its results with five clustering algorithms, k-mean, k-medoid, DB/rand/1/bin, CRO based clustering algorithm and hybrid CRO-k-mean by using four real world datasets: Lung cancer, Iris, Breast cancer Wisconsin and Haberman's survival from UCI machine learning data repository. The results were conducted and compared base on different metrics and show that proposed algorithm enhanced clustering technique by giving more accurate results.

Keywords: clustering, chemical reaction optimization, K-mean, K-medoid

1. Introduction

Clustering is the process of dividing a set of data into groups where each one called cluster. The data within the same cluster has similar characteristics and are useful, meaningful or both. If dividing the data to a meaningful group is the goal, then the data should have the same natural structure, i.e. high intra-cluster similarity and low inter-cluster similarity.

Clustering play an important role in many fields such that information retrieval, data mining, machine learning and more. Due to its important in different applications, many algorithms had been proposed before for clustering such that K-MEAN (Hartigan & Wong, 1979) , K-MEDOIDS (Chu et al., 2003), CLARANS (Chu et al., 2002), BIRCH (Raymond & Han, 2002), CURE (Mumtaz1 & Duraiswamy, 2010), DBSCAN (Zhang, Ramakrishnan & Livny, 1996), OPTICS (Alsabti, Ranka & Singh, 1998), STING (Ester et al., 1996) and CLIQUE (Matheus, Chan & Piatetsky-Shapiro, 1993). These techniques can be classified into three groups: hierarchical partitioning, and density-based. The main aim for all of them is to deal with a large set of data and classify them. Hierarchical clustering proceeds by either partition a large cluster into a set of smaller one or merge a set of clusters into a large one and then generate a hierarchy of partitions. Partition cluster decompose data into a set of disjoint clusters. Density-based create cluster of arbitrary shapes and with noise.

K-mean and k-medoid are the most prominent methods used for clustering. K-mean finds the mean of the group, called it centroid and design the prototype of object where each object belongs to the cluster with the nearest mean. K-medoid is a variant of k-mean that cluster a data on *n* objects to *k* clusters. K is the number of clusters required and is given by the user. In order to improve the performance of k-medoid algorithm, many and various methods had been proposed. In (Sheng & Liu, 2006) a hybrid Genetic algorithm with k-medoid for clustering has been proposed, the new algorithm evolve appropriate partitioning while making no **a priori** assumption about the number of clusters present in the datasets. In (Archna, Pramod & Nair, 2010), the authors proposed an enhanced version for k-medoid that eliminate deficiency of the old one by calculating the initial medoid k as per needs of users and applies an effective approach for allocation of data points into the clusters. (Swarndeep & Sharnil, 2016) proposed a new Modified K-Medoid Algorithm for improving efficiency and scalability for the study of large datasets using manthaan distance instead of Euclidean distance in terms of execution time, number of clusters and quality of clusters. In (Mariá et al., 2012), a hybrid Lagrangian heuristic for the k-medoids has been proposed and a local search method (Partition Around Medoids, PAM). The results show a very good efficiency of the heuristic considering the objective function as the parameter of comparison. In (Raghuvira et al., 2011), authors proposed an efficient density based k-medoids clustering algorithm to overcome the drawbacks of DBSCAN and k-medoids clustering algorithms. The results show that proposed Density based K-medoids algorithm performed very well than DBSCAN and k-medoids clustering in term of quality of classification measured by Rand index.

Selecting the medoid of the cluster can be improved by using an optimization method. Optimization algorithms such that Genetic (Goldberg, 1989), Chemical Reaction Optimization (CRO) (Albert et al., 2013), Particle swarm optimization (PSO) (Kennedy & Eberhart, 2011), ant colony optimization (ACO) (Socha & Dorigo, 2008), bee colony optimization (BCO) (Pham et al., 2005) and more are all used to solve large problems by finding global solution to it. Any of these optimization techniques can be used in the clustering mechanism. In this paper, the CRO meta-heuristic algorithm is used along with k-medoid to expand searching for the optimal medoid and enhance clustering by getting more precise results.

The rest of this paper is organized as follows; Section 2 give a brief overview about of K-medoid, CRO algorithms, k-mean and Differential Evolution (DE) based clustering algorithm; Section 3 proposed the work; Section 4 introduces experimental results and Section 5 gives conclusion.

2. Overview

2.1 K-medoid Clustering Algorithm

k-medoid is a classical clustering technique that partition a set of *n* objects into *k* number of clusters. Where *k* is given by the user. The principle of the algorithm is to minimize the sum of dissimilarity between each object and its corresponding reference point. It starts by randomly choose k objects from the dataset as an initial representation which is called medoids. The average of dissimilarity between the medoid and the objects in the cluster is minimal. Then for all other objects in the dataset, it assigns it to the nearest cluster depending on the distance of the object and medoid. After that, a new medoid is selected.

Input:

k: the number of clusters.

D: a data set containing n objects.

Output: A set of k clusters.

Algorithm

```
1. Select k objects from dataset D randomly as an intimal representative
2. For all the objects in D
    a. use the dissimilarity measure to find cluster C for the object i
    b. assign object i to the cluster C
    c. Randomly select a non-medoid data item and compute the total cost of swapping old medoid data item with the currently selected non-medoid data item.
    d. If the total cost of swapping is less than zero, then perform the swap operation to generate the new set of k-medoids.
```

By repeating the above steps, the algorithm tries to make better choice of medoids.

K-medoid clustering algorithm drawbacks

The k-medoid algorithm is simple but still has some drawbacks like:

- The algorithm depends on the initial random selection, using another initial selection will generate different results.
- The optimal number of k which is number of clusters is hard to be defined.
- The algorithm is sensitive to the order of data in the input dataset.

2.2 Chemical Reaction Optimization Algorithm

Chemical Reaction Optimization (CRO) is a meta-heuristic algorithm inspired by the nature of chemical reactions (Albert et al., 2013). It starts with initial molecules, and by doing a sequence of collisions, the final product become in the stable state.

The major difference between CRO algorithm and any other evolutionary techniques is that population size in the CRO may change after each iteration of the algorithm running, while in all other techniques, the population size is fixed and unchanged during running. The basic unit in the CRO population is called molecule which has a potential energy that is considered fitness function of the individual. Each molecule has a set of parameters like kinetic energy, molecule structure and more where some of them are important and some are less important depending on the problem.

In order to change on the molecule, a collision is made which could be either uni-molecule (one molecule) or inter-molecule (two or more collide with each other). The aim is to transform into a stable product with minimal potential energy.

The chemical reaction could be one of these types:

a. On-wall ineffective collision; when the molecule collides with the wall of container and then bounces. The transformation of the molecule structure can be represented as ω → ω-

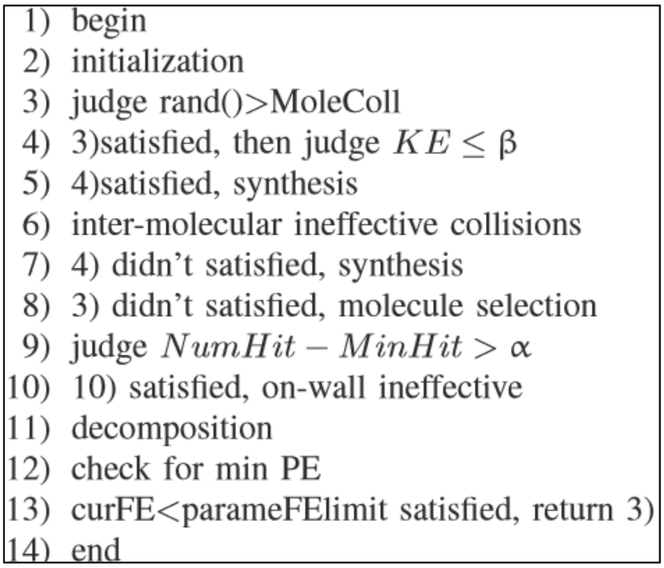
b. Decomposition; this happens when the molecule hits in the wall and then decompose into small parts. ω → ω1- + ω2

c. Inter-molecular ineffective collision; when multi molecule collide with each other and then bounces away.

d. Synthesis; which is the opposite to the decomposition and happens when two or more molecules hit and combine together. ω1 + ω2 → ω-.

The steps of the algorithm can be summarized as follow and the pseudo code is shown in Figure 1.

1. Starts with the initial population which consists of a set of individuals where each one has a potential energy (PE). Some of the CRO parameters should be defined initially such that population size, number of iterations and buffer.
2. Apply chemical reaction to generate new reactants.
3. Update the potential energy.
4. Repeat steps till reaching termination condition.



```
1) begin
2) initialization
3) judge rand()>MoleColl
4) 3)satisfied, then judge $KE \leq \beta$
5) 4)satisfied, synthesis
6) inter-molecular ineffective collisions
7) 4) didn't satisfied, synthesis
8) 3) didn't satisfied, molecule selection
9) judge $NumHit - MinHit > \alpha$
10) 10) satisfied, on-wall ineffective
11) decomposition
12) check for min PE
13) curFE<parameFElimit satisfied, return 3)
14) end
```

Figure 1. Pseudo code of the CRO algorithm

2.3 K-mean Clustering Algorithm

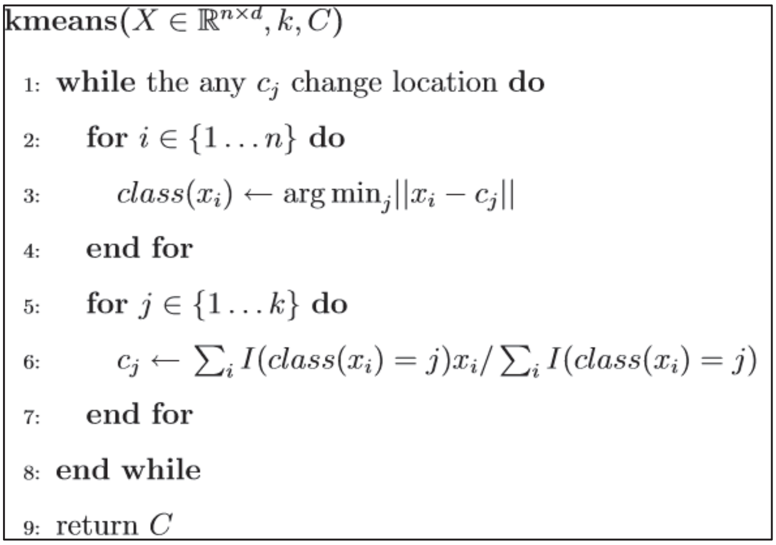
K-mean is a partitional clustering algorithm that uses a k number of clusters to partition the data. Each cluster is associate a centroid which is the center point. For each point we calculate the distance between it and the centroid, then assign it to the cluster with the minimal distance between it and its centroid. The number of centroids k must be defined previously and chosen randomly.

The main aim of the algorithm is to partition n data point into k clusters and assigning each point into disjoint cluster by minimizing the sum square error which is given in formula 1.

$$E = \sum_{i=1}^{K} \sum_{j=1}^{n} \left\| X_j - C_i \right\|^2 \quad (1)$$

Where ‖.‖ is the Euclidean distance.

The pseudo code of the algorithm is shown in Figure 2.



```
kmeans($X \in \mathbb{R}^{n \times d}, k, C$)
while the any $c_j$ change location do
    for $i \in \{1 \ldots n\}$ do
        $class(x_i) \leftarrow \arg\min_j ||x_i - c_j||$
    end for
    for $j \in \{1 \ldots k\}$ do
        $c_j \leftarrow \sum_i I(class(x_i) = j)x_i / \sum_i I(class(x_i) = j)$
    end for
end while
return $C$
```

Figure 2. Pseudo code of k-mean algorithm (Gregory, 2003)

2.4 Differential Evolution Based Clustering Algorithm

Differential evolution (DE) is an evolutionary algorithm introduced by Storn and Price in 1996 (Storn & Price, 1997), to optimize real-valued functions and real parameters. It is used with many practical problems where objective function is nondifferentiable, non-continuous, non-linear, noisy, flat, multi-dimensional or have many local minima, constraints or stochasticity in order to find an approximate solution to it.

DE is a population based, where each individual i in the population has a components d (dimensions) at time-step t (generation). The individual is changing over different generations by applying Crossover and Mutation operations based on the DE variant selected, as the DE has a wide variant developed. The old solution is replaced with the new one if the generated one is better.

DE Clustering algorithm (Panigrahi & Kumar, 2014):

1. Initialize each individual in the solution such that it contains a random selected k points from the dataset.
2. Find the fitness function for each individual.
3. Perform Crossover and Mutation operations.
4. If the generated individual has better fitness function, then replace old one with the new one.
5. Repeat previous steps till reaching termination condition.

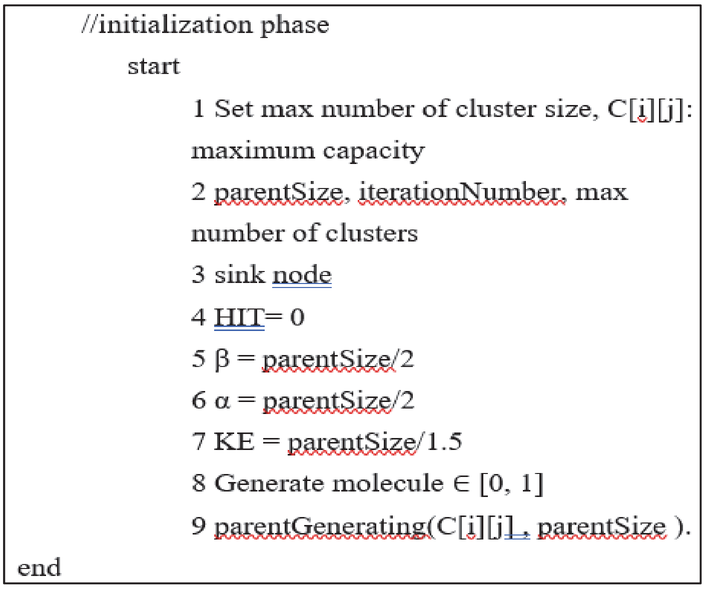
3. Proposed Work

Through our proposed idea, we will improve k-medoid clustering algorithm by using CRO meta-heuristic algorithm to select cluster medoids rather than selecting it randomly. CRO algorithm will generate better solutions by using objective function and best solution or medoid for cluster will be selected with highest objective function.

The CRO-K-medoid algorithm can be described as follow:

- **CRO-K-medoid Clustering Initialization phase**

As in (Albert et al., 2013), (Baral & Behera, 2013) CRO meta heuristic algorithm have three main phases, first phase generate a random solution as an input to the algorithm, the solution consists of individuals which represent medoid for out algorithm, each one has an objective function. The objective function for each cluster medoid must be calculated, and both inter cluster value and intra cluster value for each cluster must be calculated to compare these result for the next generation of clusters based on CRO algorithm and select best between them. In this phase also, different thresholds and variables must be defined here as we will show in the initialization generation pseudo-code phase below.



```
    //initialization phase
        start
            1 Set max number of cluster size, C[i][j]:
            maximum capacity
            2 parentSize, iterationNumber, max
            number of clusters
            3 sink node
            4 HIT= 0
            5 $\beta$ = parentSize/2
            6 $\alpha$ = parentSize/2
            7 KE = parentSize/1.5
            8 Generate molecule $\in$ [0, 1]
            9 parentGenerating(C[i][j] , parentSize ).
end
```

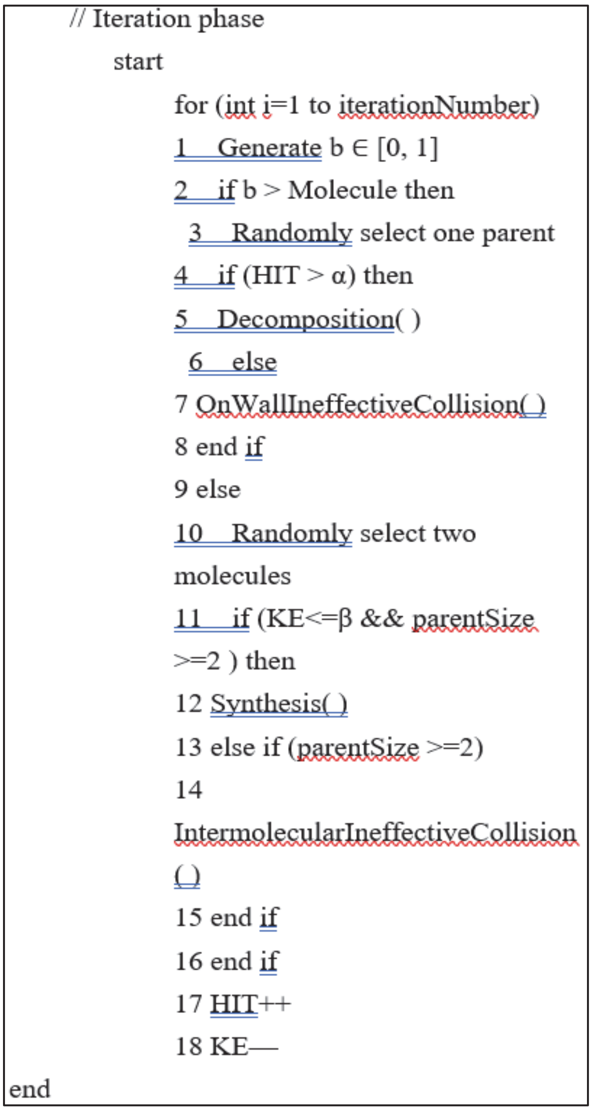
Figure 3. Pseudo code of CRO initialization phase

- **CRO-K-medoid iteration phase**

The main goal for this phase is to improve the generated first solution which selected through initialization phase. Potential energy for initial solution will be calculated here, and compared with different other solutions potential energy, where each solution will be introduced from each iteration.

Through this stage different molecule will be selected based on the value of variable “B”, where the value of variable “B” will be selected randomly between “0” and “1”. As we show from lower pseudo-code at line number 1, if the value of the variable B is larger than the value of the variable “molecule” then one molecule will be selected, this will lead to specific kind of interactions between different molecules, else if the value of variable “B” is less than the value of variable “molecule” then one molecule must be selected and this will lead to other specific interactions between molecules as decomposition and on-wall-effective collision as we show from lower molecules, each of these interactions will generate different molecule , each molecule present a specific solution with potential energy, these produced molecules must be compared with original molecules to show which solution is best than other based on potential energy value or objective function.

The pseudo-code for second phase of CRO k-medoid clustering algorithm as follow



```
// Iteration phase
start
    for (int i=1 to iterationNumber)
        Generate b ∈ [0, 1]
        if b > Molecule then
            Randomly select one parent
            if (HIT > α) then
                Decomposition( )
            else
                OnWallIneffectiveCollision( )
            end if
        else
            Randomly select two molecules
            if (KE<=β && parentSize >=2 ) then
                Synthesis( )
            else if (parentSize >=2)
                IntermolecularIneffectiveCollision( )
            end if
        end if
        HIT++
        KE—
end
```

Figure 4. Pseudo code of CRO iteration phase

- **CRO-K-medoid clustering algorithm reaction phase**

  which refer to final stage or phase of CRO-K-medoid algorithm after finishing a predefined number of iterations which present the stop criteria of our proposed algorithm, best solution here or molecule which have higher objective function value or potential energy value, as we say potential energy refer to divide both inter cluster value over intra cluster value, both inter cluster and intra cluster refer to evaluation criteria for cluster algorithm.

  In this phase we just select best solution for a set of solutions which have highest objective function value.

4. Experimental and Simulation Results

In order to evaluate performance of the proposed algorithm, the k-medoid with CRO algorithm is compared with k-mean, k-medoid, DB/rand/1/bin, CRO based clustering algorithm (Baral & Behera, 2013) and hybrid CRO-k-mean (Panigrahi & Kumar, 2014) by using four real world datasets: Lung cancer, Iris, Breast cancer Wisconsin and Haberman's survival from UCI machine learning data repository.

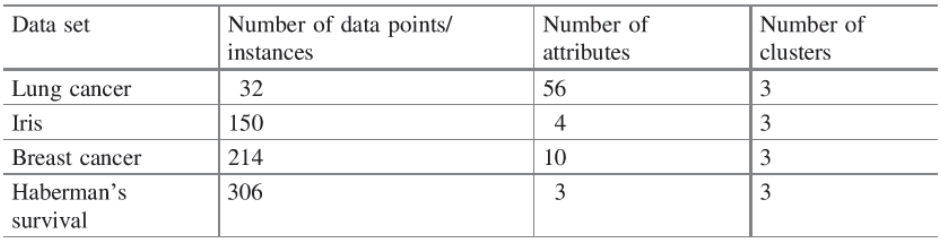
4.1 Datasets

1. Lung cancer: dataset contains 32 instances which are nominal and has an integer value from 0 to 3.
2. Iris: contains 150 patterns of three species of iris flower: setosa, versicolor and virginica.

3. Breast cancer Wisconsin: contains 699 patterns which are collected periodically by Dr. Wolberg from his clinical reports.
4. Haberman's survival: it is a study on the survival of patients who had a surgery of breast cancer between the time from 1958 to 1970 at the University of Chicago's Billings Hospital.

The attributes of each dataset that are considered is shown in table 1.

Table 1. Ddatasets attributes



Data set	Number of data points/ instances	Number of attributes	Number of clusters
Lung cancer	32	56	3
Iris	150	4	3
Breast cancer	214	10	3
Haberman's survival	306	3	3

4.2 Performance Measure

To compare performance between proposed algorithm and other algorithms, different performance measures were used as described below.

1. Clustering metric: for k clusters C1, C2, C3,….Ck, the clustering metric M can be defined as in formula 2 and should be minimize.

$$M(C_1, C_2, \ldots, C_K) = \sum_{i=1}^{K} \sum_{x_j \in C_i} ||x_j - C_i|| \tag{2}$$

2. Intra-cluster distance: is the mean of maximum distance between two data vectors within a set of clusters and should be minimized. It is given by formula 3.

$$\frac{1}{K} \sum_{i=1}^{K} \max_{ZpZq \in Ci} d(Zp, Zq) \tag{3}$$

3. Inter-cluster distance: the minimum distance between the centroids of the clusters. And should be maximized.

$$Inter\_Cluster\_Dist = \min_{j = i+1, \ldots, k} \left(\left| \mu_i - \mu_j \right|^2 \right), \quad i = 1, 2, \ldots, k-1 \tag{4}$$

4. AIC score: which is one of the metrics we used through our evaluation between both clustering algorithm K-medoid clustering and CRO-K-medoid clustering

AIC = -2 * l + 2 * k (5)

Where l refers to log of likelihoodsum(clusters), cluster here refer to number of clusters.

K refer to number of free parameters.

5. Gamma Measure: It is the index of correlation between two vectors of data with the same size. One vector is the set of distances between pairs of points and the second vector is a binary. Its range is [-1,1]. It can be described as

$$\text{Gamma } (\Gamma) = \frac{\sum_{c\{k\} \epsilon C} \sum_{M_i, M_j \epsilon c_k} d(M_i, M_j)}{N_W\left(\binom{N}{2} - N_W\right)} \tag{6}$$

Where $d(M_i,M_j)$ is the number of all point pairs in M such that M_v and M_b satisfy two conditions:

1. M_v and M_b are in different clusters and,
2. $d(M_v, M_b) < d(M_i, M_j)$.

6. G Plus Score: Other metric for clustering evaluation, which can be calculated based on this equation

$$gPlus = (2 * sMin) / (nd * (nd - 1)) \qquad (7)$$

where S min it refers to number of time a distance between two points do not belonging to the same cluster is greater than the distance between two points belonging to the same cluster (Hartigan & Wong, 1979), for that this measure should be minimized to have better cluster.

7. Point biserial Score:

Other metric measure used to evaluate clustering algorithm, through this measure it depends on calculating correlation between continuous variables and binary variables and defined as this

$$\text{Point Biserial score} = Cov\ (x,y) / Sd(x) * Sd(y) \qquad (8)$$

Where Cov () is represent covariance between two variable and sd() present the standard deviation of variables.

This value must be maximized to have a good cluster algorithm.

8. Tau Score: Which it refers to other evaluation measure to can evaluate different clustering algorithm, this evaluation measure based on this equation

$$\text{Tau} = \frac{s^{+} - s^{-}}{\sqrt{N_B N_W \left(\frac{N_T(N_T-1)}{2}\right)}} \qquad (9)$$

Where value of S- same as value which used on G-Plus calculation, S+ here refer to number of time the distance between two points which not belonging to same cluster is smaller than the distance between any two points which belonging to same cluster, based on that this measure must be maximized, the algorithm which have this value greater than other algorithm, then we can say that this algorithm is better than other algorithms.

9. Hybrid Pairwise Similarities score: This refer to measure the amount of similarity between two document which assigened to each cluster weighted according to the size of each cluster specially if we use the cosine function to measure the similarity between both documents which belong to same cluster, this value must be maximum for good clustering algorithm, which mean the similarity between different document at same cluster is high, this document similarity can be measure using different similarity measure methods as cosine method and others.

4.3 Simulation Results

Our idea for clustering using CRO meta heuristic algorithm was implemented using different data sets and compared with original version of k-medoid clustering algorithm and with 4 other clustering algorithms; k-means, DE Based Clustering Algorithm, CRO clustering algorithm and CRO-k-mean algorithm. Different criteria used for the comparison, the main two criteria which used for comparing our proposed idea with different other clustering algorithms is intra cluster distance and inter cluster distance which we used for calculating objective function for CRO algorithm.

Our proposed idea was implemented using java programing language and using cluster evaluation package on net beans 8.1, our comparison was build based on package output and based on inter cluster and intra cluster values to compare our proposed idea with different other algorithms as shown in table 2.

Table 2. Results for implementing IRIS data sets using proposed idea and other clustering algorithm

Algorithm name	Mean of Intra value	Mean of Inter value
K-means	2.59	1.59
DE-Clustering	2.56	1.79
CRO-Clustering	2.51	1.8
CRO-K-means	2.46	1.84
K-medoid	2.5	1.8026
Proposed Idea	2.513	6.08

Table1 presents results for implementing our proposed idea for clustering with other clustering algorithms on IRIS data set using same cluster number and same number of iterations, two main evaluation criteria used, inter cluster distance and intra cluster distance. inter cluster distance must be maximized as possible and intra cluster distance must be minimized as possible, upper results show that our proposed idea is the best cluster algorithm for inter cluster criteria and best all other algorithms. And regarding intra criteria it gives the second-best value.

Other 13 criteria used for cluster evaluation, each of these have a specific meaning on clustering algorithm, some of these criteria must be maximum as possible and some other must be minimum as possible.

Table 3. Different cluster evaluation criteria for evaluating both k-medoid clustering algorithm and our proposed idea

Evaluation metric name	k-medoid Result	CRO-K-medoid	Result
WB score	0.8096	0.359866	CRO Better
Trace Scatter Matrix score	149.982	149.72	CRO Better
Tau score	-3.36E-05	-2.32E-06	CRO Better
Sum Of Average Pairwise Similarities score	143.8506157	149.1769307	K-medoid Better
Point Biserial score	0.166867248	1.029402202	CRO Better
Min Max Cut score	58.2747645	51.7803872	CRO Better
Hybrid Pairwise Similarities score	0.959116486	0.996372123	CRO Better
Hybrid Centroid Similarity score	0.728994922	0.739922131	CRO Better
Gamma score	-0.893682589	-0.558334761	CRO Better
GPlus score	7.87E-05	7.28E-05	CRO Better
CIndex score	2357.99587	1493.624175	CRO Better
BIC Score score	1.01E+04	9.96E+03	K-medoid Better
AIC Score score	1.01E+04	9.96E+03	CRO Better
AVG inter distance	1.8026	2.5	CRO Better
AVG intra distance	6.08	2.513	CRO Better

Table 3 shows how our proposed idea which depends mainly on original k-medoid clustering algorithm improves original algorithm based on different evaluation metrics.

Other experiments were implemented on other data sets to show the behavior of our proposed idea as follow

Table 4. Results for implementing Lung Cancer data sets using proposed idea and other clustering algorithm

Algorithm name	Mean of Intra value	Mean of Inter value
K-means	7.51	3.18
DE-Clustering	7.76	3.27
CRO-Clustering	7.49	3.38
CRO-K-means	7.38	3.46
K-medoid	9.2549	16.538
Proposed Idea	7.397	30.19

Results in table 4 show that our proposed idea achieved better results on inter cluster distance over all different clustering algorithm including original k-means clustering algorithm with large difference on results, where intra cluster results our proposed idea achieve better results over most of clustering algorithm and achieve close result with CRO-K-means algorithm for intra cluster distance which must be minimized as possible.

Table 4 contains clustering evaluation criteria to compare our proposed idea with original k-medoid clustering algorithm as follow.

Table 5. Comparison results based on standard evaluation criteria for comparing both k-medoid clustering with CRO-Kmedoid algorithm for Lung Cancer data set

Evaluation metric name	k-medoid Result	CRO-K-medoid	Result
WB score	0.935728315	0.882217714	CRO Better
TraceScatterMatrix score	31.99445933	31.99321916	CRO Better
Tau score	-1.25E-03	-8.25E-04	K-medoid Better
Sum Of Average Pairwise Similarities score	29.38892851	29.38971474	K-medoid Better
Point Biserial score	0.223647944	0.481917834	CRO Better
Min Max Cut score	12.13700927	21.62768132	K-medoid Better

Hybrid Pairwise Similarities score	0.918563061	0.918623243	CRO Better
Hybrid Centroid Similarity score	0.819686266	0.819788632	CRO Better
Gamma score	-1	-0.848888889	CRO Better
GPlus score	2.60E-03	1.69E-03	CRO Better
CIndex score	188.6775222	346.3276152	K-medoid Better
BIC Score	2.48E+05	2.48E+05	CRO Better
AIC Score	2.48E+05	2.48E+05	K-medoid Better
AVG inter distance	16.538	30.19	CRO Better
AVG intra distance	9.2549	7.397	CRO Better

Table 5 contains different evaluation criteria results to compare both k-medoid clustering with CRO-K-medoid, our proposed idea achieves better results for 8 criteria from java package which specified for clustering evaluation and better for both inter cluster distance and intra cluster distance.

Other experiment was implemented using Breast Cancer data set which specify for clustering, same evaluation was done, first comparison between different clustering algorithm based on both inter cluster distance and intra cluster distance as follow.

Table 6. Evaluation criteria results for comparing k-medoid clustering algorithm and our proposed idea

Evaluation metric name	k-medoid Result	CRO-K-medoid	Result
WB score	0.007643387	0.023163331	K-medoid Better
TraceScatterMatrix score	197.9990709	197.9999581	K-medoid Better
Tau score	4.01E-05	2.58E-05	K-medoid Better
SumOfAveragePairwiseSimilarities score	197.9425452	197.9281711	CRO Better
Point Biserial score	1.213171056	4.00306144	CRO Better
Min Max Cut score	1758.873818	1590.910795	CRO Better
Hybrid Pairwise Similarities score	0.999714515	0.999637439	K-medoid Better
Hybrid Centroid Similarity score	0.169770962	0.169973595	CRO Better
Gamma score	0.296758105	0.920078355	CRO Better
GPlus score	1.41E-05	1.07E-06	CRO Better
CIndex score	555.9765228	2773.971728	K-medoid Better
BIC Score score	6.53E+06	6.61E+06	CRO Better
AIC Score score	6.53E+06	6.61E+06	K-medoid Better
AVG inter distance	4.42	5.3	CRO Better
AVG intra distance	15.3	15.71	CRO Better

Based on upper experimental results we can note that our proposed idea improves clustering accuracy and correctness based on different clustering evaluation criteria, upper result shows that proposed idea improves more than 9 evaluation criteria on the other hand original k-medoid clustering algorithm and show better result for evaluation criteria for 6 criteria from all 15 evaluation criteria.

Table 7. inter and intra cluster evaluation criteria results for different clustering algorithm

Algorithm name	Mean of Intra value	Mean of Inter value
K-means	17.24	3.39
DE-Clustering	16.64	4.38
CRO-Clustering	16.61	4.37
CRO-K-means	15.92	4.47
K-medoid	15.3	4.42
Proposed Idea	15.71	5.3

From upper table we can show that our proposed idea achieves higher value for inter cluster distance and second higher value for intra cluster distance.

5. Conclusions

This paper proposed a hybrid method for clustering by using chemical reaction optimization algorithm with k-medoid. The aim of the proposed method is to enhance performance of the k-medoid by expanding search for the medoid using CRO which helps on getting optimal solution, best medoid. The new algorithm was tested on different data sets and compared with original version of k-medoid algorithm and with other clustering algorithms. The proposed algorithm shows a better performance than with k-mean, k-medoid, DE, CRO clustering algorithm and CRO-K-mean.

References

Albert, Y., Lam, S., & Victor, O., Li, K. (2013). Chemical reaction optimization: a tutorial. *Memetic Computing, 4*(1).

Alsabti, K., Ranka, S., & Singh, V. (1998). An Efficient k-means Clustering Algorithm. Proc. First Workshop High Performance Data Mining, Mar. 1998.

Archna, K., Pramod, S., & Nair, S. K. (2010). An Enhanced K-Medoid Clustering Algorithm. *International Journal on Recent and Innovation Trends in Computing and Communication, 4*(6). ISSN: 2321-8169.

Baral, A., & Behera, H. S. (2013). A novel chemical reaction based clustering and its performance analysis", *Int. J. Bus. Intell. Data Mining, 8*, 184-198.

Chu, S. C., John, F., Roddick, J. F., & Pan, J. S. (2002). An Efficient K -MedoidsBased Algorithm Using Previous Medoid Index, Triangular Inequality Elimination Criteria, and Partial Distance Search" 4th International Conference on Data Warehousing and Knowledge Discovery, 2002.

Chu, S. C., Roddick, J. F., Chen, T. Y., & Pan, J. S. (2003). Efficient search approaches for k-medoids-based algorithms. TENCON '02. Proceedings. 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering, Feb 2003.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases", Proc. 1996 Int. Conf. Knowledge Discovery and Data Mining (KDD'96), Portland, OR, August 1996.

Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. Reading, MA: Addison-Wesley

Gregory, J. H. (2003). Learning structure and concepts in data through data clustering. A dissertation submitted in partial satisfaction of the requirements for the degree Doctor of Philosophy in Computer Science and Engineering, University of California, San Diego.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society, Series C., 28*(1), 100–108.

Kennedy, J., & Eberhart, R. (2011). Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco.

Mariá, C. V., Nascimento, Franklina, M. B., Toledo, & André, C. P. L. F. (2012). A Hybrid Heuristic for the k-medoids Clustering Problem", GECCO'12, July 7-11, 2012, Philadelphia, Pennsylvania, USA. Copyright 2012 ACM 978-1-4503-1177-9/12/07.

Matheus, C. J., Chan, P. K., & Piatetsky-Shapiro, G. (1993). Systems for Knowledge Discovery in Databases, *IEEE Transactions on Knowledge and Data Engineering, 5*(6), 903-913.

Moh'd, B., Al-Zoubi, Amjad, H, Ammar, H., & Bassam, H. (2008). New Efficient Strategy to Accelerate k-Means Clustering Algorithm. *American Journal of Applied Sciences, 5*(9), 1247-1250.

Moh'd, B., Al-Zoubi, Amjad, H, Ammar, H., & Al-Shboul, B. (2007). A Fast Fuzzy Clustering Algorithm. Proceedings of the 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, Corfu Island, Greece, 16-19, 28.

Mumtaz1, K., & Duraiswamy, D. K. (2010). A Novel Density based improved k-means Clustering Algorithm – Dbkmeans. *International Journal on Computer Science and Engineering, 2*(2), 213-218.

Panigrahi, P. R., & Kumar, P. S. (2014). A Hybrid CRO-K-Means Algorithm for Data Clustering. *Computational Intelligence in Data Mining, 3*, 627-639.

Pham, D. T., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S., & Zaidi, M. (2005). The Bees Algorithm – A Novel Tool for Complex Optimisation Problems", Technical Note: MEC 0501. UK: The Manufacturing Engineering Centre, Cardiff University, Cardiff, UK, Queen's University, Belfast, UK.

Raghuvira, P. A., Suvarna, K., Vani, J., Rama, D. K., & Nageswara, R. (2011). An Efficient Density based Improved K- Medoids Clustering algorithm. *(IJACSA) International Journal of Advanced Computer Science and Applications, 2*(6).

Raymond, T. N., & Han, J. W. (2002). CLARANS: A Method for Clustering Objects for Spatial Data Mining, *IEEE Transactions on Knowledge and Data Engineering, 14*(5), SEPTEMBER 2002.

Sheng, W. G., & Liu, X. H. (2006). A genetic k-medoids clustering algorithm", Submitted in December 2004 and accepted by Zbigniew Michalewicz in January 2005 after 2 revisions C Springer Science+Business Media, LLC.

Socha, K., & Dorigo, M. (2008). Ant colony optimization for continuous domain. *European Journal of Operational Research, 185*, 1155-1173.

Storn, R., & Price, K. (1997). Differential evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization, 11*(4), 341-359.

Swarndeep, S. J., & Sharnil, P. (July 2016). Implementation of Extended K-Medoids Algorithm to Increase Efficiency and Scalability using Large Datasets. *International Journal of Computer Applications (0975 – 8887), 146*(5).

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases, In: SIGMOD Conference, pp.103-114.

Copyrights