
An Improved Cat Swarm Optimization Algorithm Based on Opposition-Based Learning and Cauchy Operator for Clustering

Yugal Kumar* and G. Sahoo*

Abstract

Clustering is a NP-hard problem that is used to find the relationship between patterns in a given set of patterns. It is an unsupervised technique that is applied to obtain the optimal cluster centers, especially in partitioned based clustering algorithms. On the other hand, cat swarm optimization (CSO) is a new meta-heuristic algorithm that has been applied to solve various optimization problems and it provides better results in comparison to other similar types of algorithms. However, this algorithm suffers from diversity and local optima problems. To overcome these problems, we are proposing an improved version of the CSO algorithm by using opposition-based learning and the Cauchy mutation operator. We applied the opposition-based learning method to enhance the diversity of the CSO algorithm and we used the Cauchy mutation operator to prevent the CSO algorithm from trapping in local optima. The performance of our proposed algorithm was tested with several artificial and real datasets and compared with existing methods like K-means, particle swarm optimization, and CSO. The experimental results show the applicability of our proposed method.

Keywords

Cat Swarm Optimization, Cauchy Mutation Operator, Clustering, Opposition-Based Learning, Particle Swarm Optimization

1. Introduction

Clustering is popular technique that has been applied to many research domains like image analysis, pattern recognition, data mining, medical science, etc. The aim of clustering is to find groups of similar objects, which known as clusters, and objects within clusters that share similar characteristics. A similarity criterion function is used to group the objects into clusters. In literature, various similarity criterion functions are defined, but the popular one is the Euclidean distance. It is also noticed that the clustering problem has shown exponential complexity in terms of the number of clusters and it becomes an NP-hard problem when the number of clusters increases to more than three. In clustering, K-means is one of the oldest and most well-known algorithms and this algorithm is applied to obtain optimal cluster centers [1]. But, it suffers from several shortcomings, like trapping in local optima, the

* This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Manuscript received February 5, 2014; first revision January 6, 2015; accepted March 30, 2015; onlinefirst June 2, 2015.

Corresponding Author: Yugal Kumar (yugalkumar.14@gmail.com, yugalkumar@bitmesra.ac.in)

* Dept. of Computer Science and Engineering, Birla Institute of Technology, Mesra, Ranchi, Jharkhand, India (gsahoo@bitmesra.ac.in)

fact that the quality of the solution depends on the initial cluster centers, and there being a lack of information to treat with inappropriate and clutter attributes [2,3]. From the literature, it came to revelation that heuristic approaches are more suitable and popular for solving the clustering problem. Several heuristic approaches have been applied to obtain optimal solutions for clustering problems. Over the past few decades, a number of evolutionary algorithms have been designed to solve clustering problems. These algorithms have the ability to change themselves according to the problem description and can be applied to solve a variety of optimization problems using only a few changes. These algorithms also handle constraints in an effective way. To investigate the potential of the EA's technique for solving the clustering problem, it has been applied to obtain the optimal cluster centers [4,5]. The particle swarm optimization (PSO) algorithm has been applied to obtain the optimal cluster centers [6]. Shelokar et al. [7] have applied the ant colony optimization (ACO) method for effective clustering. A teaching-learning-based optimization (TLBO) method has been discussed for data clustering in [8]. Data clustering based on the behavior of charged particles is presented in [9,10]. An artificial bee colony (ABC) algorithm for data clustering has been described in [11]. Application of the tabu search method for clustering is reported in [12]. Apart from these, hybrid versions of these algorithms are also reported for clustering problems. A genetic K-means algorithm has been presented for data clustering. This is where a genetic algorithm is utilized to compute the initial cluster centers for the K-means algorithm [13]. Hybridization of the PSO with other algorithms has also been developed by numerous researchers to solve the clustering problems [14-16]. Hybrid versions of ABC, GA, TLBO, and ACO algorithms are also found in [17-22]. Still, there are some shortcomings associated with these algorithms, such as trapping in local optima, sensitive to initial cluster centers, boundary level constraints, and quality of solutions. A lot of efforts have been going on in this direction to overcome these shortcomings.

In this research work, we applied the cat swarm optimization (CSO) algorithm to obtain the optimal cluster centers for clustering problems. CSO is the latest meta-heuristic algorithm developed by Chu et al. [23], and is based on the behavior of cats. The CSO algorithm has been applied in various fields and has provided remarkable results [24-28]. Santosa and Ningrum [24] have applied the CSO algorithm for clustering and they measured the performance of the CSO algorithm on four datasets using classification error, mean, and standard deviation parameters. The results show that CSO provides more accurate results in terms of classification errors in comparison to K-means and PSO. But, this algorithm suffers several shortcomings, such as algorithm trapped in local optima, diversity problems, and there not being a predefined method to deal with data objects that cross the boundary constraints. Hence, to address the above mentioned shortcomings and also to improve the results of the CSO algorithm for the clustering problem, we are proposed few modifications in the original CSO algorithm. These modifications are summarized as listed below.

- The opposition-based learning method is used to enhance the population diversity of the CSO algorithm.
- A Cauchy mutation operator has been introduced in the CSO algorithm to overcome the local optima problem, as well as to enhance the diversity of the CSO algorithm.
- Some heuristic are also proposed to deal with the data vectors trapped in in local optima, if the solution lies near the boundary of the datasets.

2. Cat Swarm Optimization

Chu et al. [23] have introduced a new meta-heuristic algorithm based on the behavior of cats and named it CSO. Cats have two distinct characteristics that make them different from other species. One is a strong curiosity about moving objects and the other is outstanding hunting skills. Based on these characteristics, the two modes of the seeking mode and the tracing mode are identified [23]. The seeking mode describes the curiosity of cats in regards to moving objects, while the tracing mode describes their outstanding hunting skills. A mathematical model is formed by combining these two modes to solve the optimization problems. In d-dimensional space, cats are represented using the position and velocity vectors. A problem specific fitness function is defined to direct the next search step. In the CSO algorithm, the position of the cats represent the possible solution set and a flag is used to determine the cat's mode.

2.1 Seeking Mode

The seeking mode of the CSO algorithm can be viewed as a global search for the solution in the random search space for an optimization problem. The terms related to the seeking mode are discussed as listed below.

- Seeking Memory Pool (SMP): this can be defined as the number of copies of a cat produced in the seeking mode.
- Seeking Range of the selected Dimension (SRD): this can be defined as the maximum difference between the old and new positions of cats in the dimensions that have been selected for mutation.
- Counts of Dimension to Change (CDC): this can be defined as the number of dimensions go for mutation.

The steps involved in this mode are:

1. Make 'i' copies of cat_j , where 'i' equals the seeking memory pool. If 'i' is one of candidate solutions, then 'i'=SMP-1 else 'i'=SMP
2. Determine the shifting value for each 'i' copies using (SRD*position of cat_j).
3. Determine the number of copies undergo mutation (Randomly add or subtract the shifting value to 'i' copies).
4. Evaluate the fitness of all copies.
5. Pick the best candidate from the 'i' copies of cat_j and place it at the of j^{th} position cat.

2.2 Tracing Mode

The tracing mode of the CSO algorithm is similar to local search. In this mode, cats update their velocities and positions by targeting the objects with high speed. As a result of this, a large difference occurs between the positions of the cats. The position (X_j^d) and velocity (V_j^d) of the j^{th} cat in the d-dimensional space is described as: $X_j^d = \{X_j^1, X_j^2, \dots \dots X_j^D\}$ and $V_j^d = \{V_j^1, V_j^2, \dots \dots V_j^D\}$ where, $d = 1, 2, \dots D$. The best position of the cat is represented as: $X_{best}^d = \{X_{best}^1, X_{best}^2, \dots \dots X_{best}^D\}$ and the velocity and position of the j^{th} cat is updated using Eqs. (1) and (2).

$$V_{j_{new}}^d = w * V_j^d + c * r * (X_{j_{best}}^d - X_j^d) \quad (1)$$

where, $V_{j_new}^d$ represents the new velocity of the j^{th} cat in the d^{th} dimension, w denotes a weight factor in the range of 0 and 1, V_j^d represents the old velocity of the j^{th} cat, c is a user defined constant, r denotes a random number in the range of 0 and 1, $X_{j_best}^d$ represents the best position achieved by the j^{th} cat, X_j^d denotes the current position of the j^{th} cat, and $d = 1, 2, \dots, D$.

$$X_{j_new}^d = X_j^d + V_j^d \quad (2)$$

where, $X_{j_new}^d$ denotes the updated position of the j^{th} cat, X_j^d denotes the current position of the j^{th} cat, and V_j^d represents the velocity of the j^{th} cat

3. Improved Cat Swarm Optimization

This section describes the structure of our proposed algorithm. In order to make the CSO algorithm more effective and competent for clustering problems, a few modifications are inculcated into the original CSO algorithm. The diversity nature of the CSO algorithm is enhanced using the opposition-based learning method and the Cauchy mutation operator is used to overcome the local optima problem. The detailed description of these modifications are described below.

3.1 Cauchy Mutation Operator

The Cauchy mutation operator is used to prevent the CSO algorithm from falling into the local optima, especially in the tracing mode. Many researchers have introduced the concept of mutation [29-32]. The idea behind the inclusion of a mutation operator with heuristic approaches is to prevent the local optima and to increase the population diversity. To achieve the same, the best position of cat is mutated. The Cauchy mutation operator is explained using Eqs. (2) and (3).

$$W(d) = \sum_{d=1}^D V[j][d] / D \quad (3)$$

where, $V[j][d]$ represents the velocity vector of the j^{th} cat in the d^{th} dimension, $d = 1, 2, \dots, D$, and $W(d)$ is a weight vector in the range of $[-W_{\min}, W_{\max}]$.

$$X_{j_best,new}^d = X_{j_best}^d + W(d) * N * \min(X_{max}^d - X_{j_best}^d, X_{j_best}^d - X_{min}^d) \quad (4)$$

In Eq. (4), $X_{j_best,new}^d$ denotes the new mutated best position of the j^{th} cat, $X_{j_best}^d$ represents the best position of j^{th} cat in the d^{th} dimension, N represents the Cauchy distribution function in the range of 0 and 1. $(X_{max}^d - X_{j_best}^d, X_{j_best}^d - X_{min}^d)$ denotes the difference between the minimum and maximum values of the d^{th} dimension of a dataset and the j^{th} best position of a cat.

3.2 Opposition Based Learning

Rahnamayan et al. [33,34] introduced the concept of opposition-based learning with the meta-

heuristic approaches to solve the optimization problems. This concept is further explored by Wang et al. [35,36]. According to opposition-based learning, assume that X is the solution for a given problem; then, the opposite of X will be the other candidate solution. In this case, the chances to obtain the optimal solution will be increased.

Opposite number [35]: let $X \in (a, b)$ be a real number; then, the opposite of x is given by:

$$X' = a + b - X \tag{5}$$

In Eq. (5), X' denotes the opposite solution, X denotes the current best solution, 'a' and 'b' are two constants. Moreover, in the d -dimensional search space the above equation can be rewritten as. Consider $X = (X_1, X_2, \dots, X_d)$ is a point in the d -dimensional space, where, $(X_1, X_2, \dots, X_d) \in R$ and $X_d \in (a_d, b_d)$, $d = (1, 2, 3, \dots, D)$. The opposite number can be described as: $X' = X'_1, X'_2, \dots, X'_d$.

$$X_j^{d'} = a_j^d + b_j^d - X_j^d \tag{6}$$

The opposition-based learning can be defined as. Assume $X = (X_1, X_2, \dots, X_d)$ is a point in the d -dimensional space (i.e., a candidate solution) and $f(X)$ is a fitness function that is used to evaluate the fitness of candidate solutions. According to the above definitions $X' = X'_1, X'_2, \dots, X'_d$ is the opposite of $X = (X_1, X_2, \dots, X_d)$. If $f(X)$ is better, then update X ; otherwise, X' . It is also mentioned that both the X and X' are simultaneously computed and keep the best one. The variables a_d and b_d denote the minimum and maximum values of the j^{th} dimension.

3.3 Boundary Constraints

In the CSO algorithm, when a data instance crosses the boundary constraints of the dataset, then it is replaced with the values that are near the boundary of the dataset. If the data instances cross the boundary constraints frequently, then the algorithm assigns the values to each data instance near the dataset boundary. Therefore, there is a possibility to trap the algorithm in the local optima and lose the diverse nature. Hence, to overcome this problem, the two following modifications are proposed in the CSO method: one for the seeking mode and another for the tracing mode. In the seeking mode, the addition and subtraction of the shifting value to the cluster centers may lead the data vectors to cross the boundary of the dataset. Hence, a mechanism has been introduced to deal with such data vectors and the proposed mechanism can be defined as explained below.

If the data vector $X_j^d < X_{\min}^d$ then:

$$X_{j_{new}}^d = X_{\min}^d + [rand(0,1) * (X_{\max}^d - X_{\min}^d) + a] \tag{7}$$

where, $X_{j_{new}}^d$ is the new position of the j^{th} cat in the d^{th} dimension, r is a random number in the range of $[0,1]$, X_{\max}^d and X_{\min}^d are the maximum and minimum values of the d^{th} dimension of the dataset, and 'a' is a variable that is used to prevent the data vectors from getting stuck in the local optima near the boundary of the dataset. The variable 'a' is calculated using the following equation:

$$a = (1 + \text{current iteration} / \text{maximum iteration}) \tag{8}$$

In Eq. (8), the current iteration denotes the present iteration number and the maximum iteration defines the total number of iteration sets to execute the algorithm.

If the data vector $X_j^d > X_{max}^d$, then:

$$X_{j_{new}}^d = X_{max}^d - [rand(0,1) * (X_{max}^d - X_{min}^d) + a] \quad (9)$$

where, 'a' is a variable that is used to prevent the data vectors from being getting stuck in the local optima and it is calculated using Eq. (10):

$$a = (1 - current\ iteration / maximum\ iteration) \quad (10)$$

Another modification is employed in the tracing mode of the CSO algorithm. In the tracing mode, a cat traces his/her target with high speed. Mathematically, it can be achieved by defining the position and velocity of cats in the d-dimensional random search space. Thus, Eq. (2) obtains the new position of the j^{th} cat. As such, there is possibility that the positions of the cats may cross the boundary limits. To deal with such data vectors, another method is described and the proposed method can be outlined as listed below.

When any data vector $X_j^d < X_{min}^d$, then:

$$X_{j_{new}}^d = X_{min}^d + V_{j_{new}}^d; \quad V_{j_{new}}^d = rand(0,1) * |V_j^d| \quad (11)$$

When any data vector $X_j^d > X_{max}^d$, then:

$$X_{j_{new}}^d = X_{max}^d + V_{j_{new}}^d; \quad V_{j_{new}}^d = -rand(0,1) * |V_j^d| \quad (12)$$

Hence, as discussed above the population diversity of the CSO method is enhanced by using opposition-based learning and the local optima problem is sorted by using the Cauchy mutation operator. As a result of these modifications, the algorithm can explore more solution and it also limits the data vectors in a random search space (using Eqs. (7) and (9)), which gives better results. The flow chart of the opposition-based learning-improved CSO (OL-ICSO) is shown in Fig. 1.

3.4 Steps of the OL-ICSO Algorithm

1. Load the dataset. Initialize the parameters for the CSO method and the number of cats.
2. Initialize the positions of the cats in random fashion and evaluate the velocity of every cat.
3. Compute the opposite position (X') of the cats using Eq. (6).
4. Compute the Euclidean distance for each data from each cluster center and group the data into different clusters using the objective function values.
5. Determine the fitness function (SSE) of each cat_k and keep the best position of cat_k in variable $X_{g_{best}}$ of cat_k .
6. For each cat_k , apply the seeking mode process:
 - a. Make the j copy of each cat_k .
 - b. Calculate the shifting value for each cat_k using (SRD*cluster center (k)).
 - c. Add or subtract each cat_k to/from the shifting value.

- d. Calculate the objective function value, group, and data.
- e. Evaluate the value of fitness function (SSE) and keep the best position X_{sbest} of cat_k .
- f. If $SSE \leq SSE_s$, then $SSE_{sbest} \rightarrow SSE$ and $X_{sbest} \rightarrow X_{best}$.
- g. Else, $SSE_{sbest} \rightarrow SSE_s$ and $X_{sbest} \rightarrow X_{sbest}$
7. For each cat_k , apply the tracing mode process:
 - a. Update the velocity of cat_k using Eq. (1).
 - b. Update the position of each cat_k using Eq. (2).
 - c. Calculate the objective function value and group the data into different clusters.
 - d. Evaluate the fitness function value (SSE_{tbest}) and keep the best position X_{tbest} of cat_k .
8. If $\{SSE_{sbest} \leq SSE_{tbest}\}$, $SSE_{gbest} \rightarrow SSE_{tbest}$ and $X_{gbest} \rightarrow X_{sbest}$, else $SSE_{gbest} \rightarrow SSE_{tbest}$ and $X_{gbest} \rightarrow X_{sbest}$.
9. If $\{SSE_{gbest} \geq rand(\)\}$ then, goto Step 6.
10. Else, mutate the position X_{gbest} using Eq. (11).
11. Evaluate the fitness function ($SSE_{Mut.}$) values, keep the best position in X_{Mbest} .
12. If $\{SSE_{Mut.} \leq SSE_{gbest}\}$, then $X_{gbest} \rightarrow X_{Mbest}$ and compute the opposite of X_{Mbest} using Eq. (4).
13. Go to Step 6, until the maximum iteration is reached.
14. Obtain the final solutions.

Where, SSE_{sbest} , SSE_{tbest} , $SSE_{Mut.}$, and SSE_{gbest} represent the value of the fitness function in the seeking mode, tracing mode, the Cauchy mutation operator, and the global fitness of the CSO algorithm. X_{sbest} , X_{tbest} , X_{Mbest} , and X_{gbest} denote the best position achieved by cat_k in the seeking mode, tracing mode, the Cauchy mutation operator, and global best.

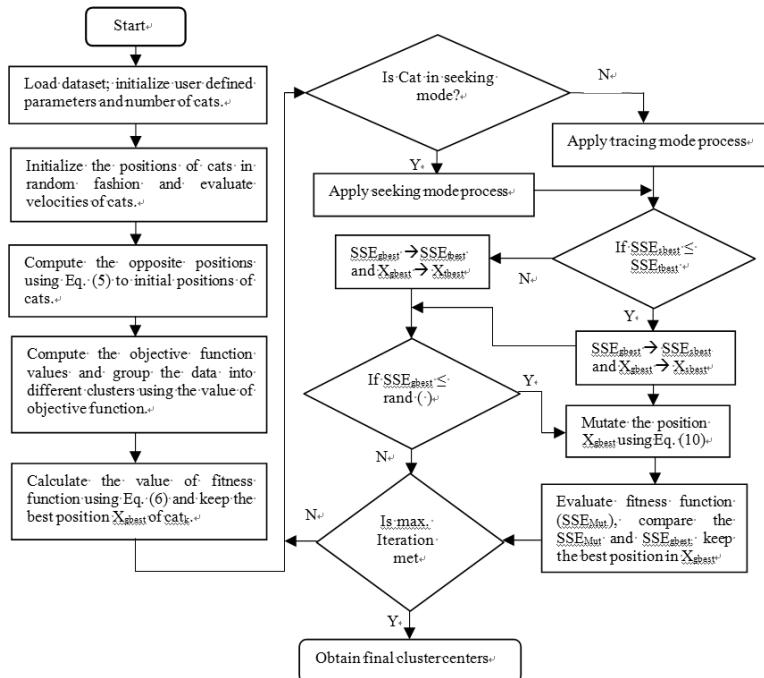


Fig. 1. Flowchart of proposed opposition-based learning improved cat swarm optimization (OL-ICSO) algorithm.

4. Experimental Results

This section describes the simulation results of the proposed algorithm with several datasets from the UCI repository. We compared the performance of the proposed algorithm against the K-Means, PSO, and CSO algorithms using the sum of the intra cluster distance, standard deviation, and F-measure parameters. We evaluated the performance of the proposed algorithm using ART1, ART2, iris, CMC, cancer, and wine datasets. The ART1 and ART2 datasets are artificial datasets that were generated in MATLAB 2010a. While, the rest of datasets are real datasets that were taken from the UCI repository. We used the MATLAB 2010a environment to implement the proposed algorithm. The algorithm was independently run 20 times with randomly initialized cluster centers.

4.1 Performance Measures

Intra Cluster Distance

Intra cluster distance can be defined as the distance between the data objects within clusters to their respective cluster centers. This parameter also indicates the quality of clustering (i.e., the smaller the intra cluster distance, the better the quality of the solution). The results are measured in terms of best, average, and worst solutions.

Standard Deviation

Standard deviation gives the information about the scattering of data within a cluster. A lower standard deviation value indicates that the data objects are scattered near cluster centers, while a high value indicates that the data is dispersed away from its center point.

F-Measure

F-Measure can be measured in terms of the recall and precision of an information retrieval system. It is also described as a weighted harmonic mean of recall and precision. The value of the F-measure, $F(i, j)$, is computed as follows:

$$F(i, j) = \frac{2 * (Recall * Precision)}{(Recall + Precision)} \quad (13)$$

The value of the F-measure for a given clustering algorithm that consists of n number of data instances is given as:

$$F(i, j) = \sum_{i=1}^n \frac{n_i}{n} * max_i * F(i, j) \quad (14)$$

4.2 Datasets

The detailed descriptions of ART1 and ART2 datasets are given as below and the characteristics of the rest of the datasets are summarized in Table 1.

ART1: This is a two-dimensional dataset that is created in MATLAB to validate the proposed algorithm. It consists of 300 instances with two attributes and three classes. Classes in the dataset are circulated using μ and λ , where μ is the mean vector, λ is the variance matrix, and the values of $\mu_1 = [3, 1]$, $\mu_2 = [0, 3]$, $\mu_3 = [1.5, 2.5]$ and $\lambda_1 = [0.3, 0.5]$, $\lambda_2 = [0.7, 0.4]$, $\lambda_3 = [0.4, 0.6]$.

ART2: This is a three-dimensional data that consists of 300 instances with three attributes and three classes. Data is created using $\mu_1 = [10, 25, 12]$, $\mu_2 = [11, 20, 15]$, $\mu_3 = [14, 15, 18]$ and $\lambda_1 = [3.4, -0.5, -1.5]$, $\lambda_2 = [-0.5, 3.2, 0.8]$, $\lambda_3 = [1.5, 0.1, 1.8]$.

Table 1. Characteristic of datasets

| Dataset | Class | Feature | Total instance | Instance in each class |
|---------|-------|---------|----------------|------------------------|
| ART1 | 3 | 2 | 300 | (100, 100, 100) |
| ART2 | 3 | 3 | 300 | (100, 100, 100) |
| Iris | 3 | 4 | 150 | (50, 50, 50) |
| Cancer | 2 | 9 | 683 | (444, 239) |
| CMC | 3 | 9 | 1473 | (629, 334, 510) |
| Wine | 3 | 13 | 178 | (59, 71, 48) |

4.3 Parameter Settings

In order to evaluate the performance of the proposed algorithm, user-defined parameters have to be processed first. The proposed algorithm consists of five user-defined parameters, such as SRD, SMP, c_1 , r_1 , and SPC. The parameter settings of the proposed algorithm and other algorithms being compared are given in Table 2.

Table 2. Parameters value for PSO, CSO, ICSO and OL-ICSO algorithms

| Parameter | PSO | CSO | ICSO |
|------------|--------|--------|--------|
| Max. Iter. | 100 | 100 | 100 |
| SRD | - | (0, 1) | (0, 1) |
| SMP | - | 5 | 5 |
| SPC | - | (0, 1) | (0, 1) |
| w | 1 | - | - |
| c_1 | 2 | 2 | 2 |
| c_2 | 2 | - | - |
| r_1 | (0, 1) | (0, 1) | (0, 1) |
| r_2 | (0, 1) | - | - |
| Vel. Max. | 0.9 | 0.9 | 0.9 |
| W_{max} | - | - | 0.5 |
| W_{min} | - | - | -0.5 |

PSO=particle swarm optimization, CSO=cat swarm optimization, ICSO=improved CSO, OL-ICSO=opposition-based learning ICSO.

Table 3. Performance comparison of different algorithms

| Dataset | Parameter | K-means | PSO | CSO | ICSO | OL-ICSO |
|---------|-----------|----------|----------|----------|----------|----------|
| ART1 | Best | 157.12 | 154.06 | 154.26 | 154.13 | 153.76 |
| | Average | 161.12 | 158.24 | 159.06 | 158.17 | 157.89 |
| | Worst | 166.08 | 161.83 | 164.56 | 162.08 | 159.24 |
| | Std. | 0.34 | 0 | 0.292 | 0.14 | 0.236 |
| | F-measure | 99.14 | 100 | 100 | 100 | 100 |
| ART2 | Best | 743 | 740.29 | 740.18 | 740.08 | 741.43 |
| | Average | 749.83 | 745.78 | 745.91 | 745.74 | 745.81 |
| | Worst | 754.28 | 749.52 | 749.38 | 748.24 | 747.56 |
| | Std. | 0.516 | 0.237 | 0.281 | 0.247 | 0.14 |
| | F-measure | 98.94 | 99.26 | 99.32 | 99.35 | 99.42 |
| Iris | Best | 97.33 | 96.89 | 96.97 | 96.78 | 96.31 |
| | Average | 106.05 | 97.23 | 97.16 | 97.08 | 96.79 |
| | Worst | 120.45 | 97.89 | 98.18 | 97.83 | 97.36 |
| | Std. | 14.631 | 0.347 | 0.192 | 0.156 | 0.12 |
| | F-measure | 0.782 | 0.782 | 0.782 | 0.783 | 0.786 |
| Cancer | Best | 2999.19 | 2973.5 | 2992.45 | 2967.07 | 2954.18 |
| | Average | 3251.21 | 3050.04 | 3109.14 | 3036.49 | 3024.06 |
| | Worst | 3521.59 | 3318.88 | 3456.63 | 3291.16 | 3217.64 |
| | Std. | 251.14 | 110.801 | 132.47 | 43.56 | 36.78 |
| | F-measure | 0.829 | 0.819 | 0.831 | 0.834 | 0.835 |
| CMC | Best | 5842.2 | 5700.98 | 5696.23 | 5685.76 | 5628.63 |
| | Average | 5893.6 | 5820.96 | 5778.12 | 5756.31 | 5741.16 |
| | Worst | 5934.43 | 5923.24 | 5908.32 | 5917.21 | 5892.24 |
| | Std. | 47.16 | 46.959 | 41.33 | 35.04 | 31.47 |
| | F-measure | 0.334 | 0.331 | 0.336 | 0.341 | 0.344 |
| Wine | Best | 16555.68 | 16345.96 | 16331.56 | 16317.46 | 16309.54 |
| | Average | 18061 | 16417.47 | 16395.18 | 16357.89 | 16348.46 |
| | Worst | 18563.12 | 16562.31 | 16548.54 | 16534.76 | 16568.87 |
| | Std. | 793.213 | 85.497 | 57.34 | 40.73 | 36.24 |
| | F-measure | 0.521 | 0.518 | 0.523 | 0.524 | 0.525 |

PSO=particle swarm optimization, CSO=cat swarm optimization, ICSO=improved CSO, OL-ICSO=opposition-based learning ICSO.

4.4 Results and Discussion

Table 3 shows the simulation results of the K-means, PSO, CSO, ICSO, and OL-ICSO algorithms. It is shown that the OL-ICSO algorithm provides better results in comparison to other algorithms. It is also noted that improvements in the CSO algorithm (ICSO and OL-ICSO algorithms) not only improves the

results with ART1 and cancer datasets but also enhances the results with all other datasets using all of the parameters. It can be observed from the results that the OL-ICSO algorithm obtains a minimum intra cluster distance using all of the datasets, as well as higher values of F-measure parameters from among all of the algorithms. Figs. 2 and 3 show the convergence of the intra-cluster distance and f-measure parameters for the wine dataset using all of the algorithms. The time consumptions of all of these algorithms with artificial datasets are nearly the same. In the case of the iris, cancer, CMC, and wine datasets, the K-means algorithm takes less execution time than the other algorithms and the time it takes to do so is 18, 26, 49, and 43 seconds, respectively. The time consumptions of the OL-ICSO algorithm for real datasets are 51, 131, 143, and 106 seconds, respectively. While, the time consumptions of the CSO algorithm are 47, 139, 148, and 97 seconds, respectively.

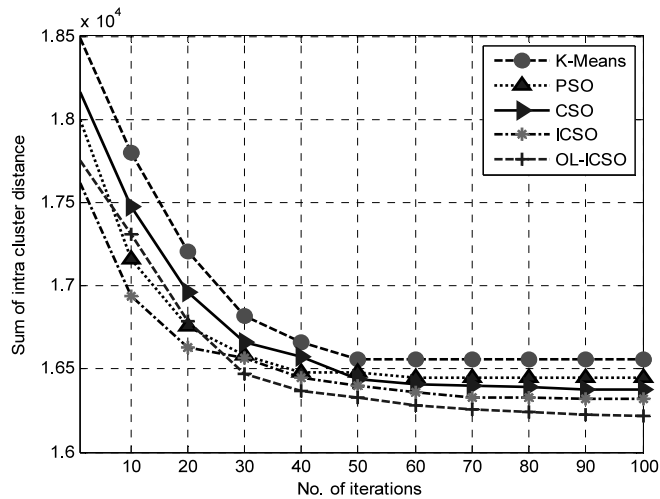


Fig. 2. Convergence of sum of intra cluster distance parameter using K-means, PSO, CSO, ICSO and OL-ICSO for wine dataset. PSO=particle swarm optimization, CSO=cat swarm optimization, CSO=improved CSO, OL-ICSO=opposition-based learning ICSO.

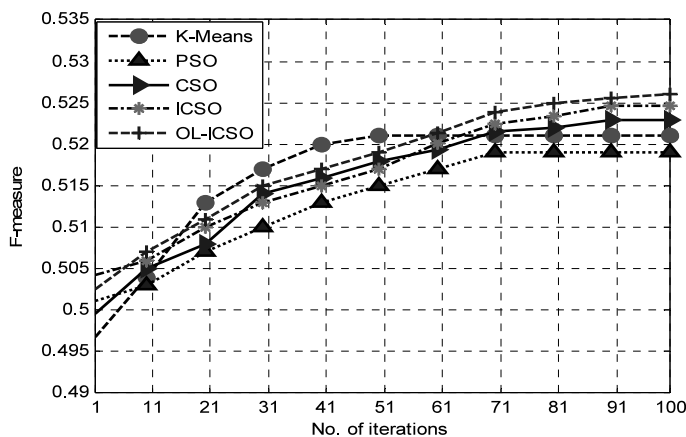


Fig. 3. Convergence of F-measure parameter using K-means, PSO, CSO, ICSO and OL-ICSO for wine dataset. PSO=particle swarm optimization, CSO=cat swarm optimization, ICSO=improved CSO, OL-ICSO=opposition-based learning ICSO.

5. Conclusion

In this work, opposition-based learning and the Cauchy mutation operator are applied to enhance the population diversity and prevent the local optima problem of the CSO algorithm. Moreover, some heuristics are also proposed to deal such data vectors which cross the boundary of the dataset. From Table 3, it is observed that the Cauchy mutation operator improves the performance of the CSO algorithm (ICSO) in comparison to the original CSO algorithm. However, the population diversity of the CSO algorithm is not improved. In order to enhance the population diversity of the CSO algorithm, the opposition-based learning concept is used in combination with the Cauchy mutation operator. From the results, it is concluded that the opposition-based learning method with Cauchy mutation operator (OL-ICSO) combination not only enhanced the population diversity, but also prevents the local optima problems of the CSO algorithm. We also discovered that the proposed OL-ICSO algorithm provides better performance in comparison to K-Means, PSO, CSO, and ICSO algorithms using six datasets.

References

- [1] J. MacQueen, "On convergence of k-means and partitions with minimum average variance," *Annals of Mathematical Statistics*, vol. 36, no. 3, p. 1084, 1965.
- [2] S. Z. Selim and M. A. Ismail, "K-means-type algorithms: a generalized convergence theorem and characterization of local optimality," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 1, pp. 81-87, 1984.
- [3] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651-666, 2010.
- [4] C. A. Murthy and N. Chowdhury, "In search of optimal clusters using genetic algorithms," *Pattern Recognition Letters*, vol. 17, no. 8, pp. 825-832, 1996.
- [5] L. Y. Tseng and S. B. Yang, "A genetic approach to the automatic clustering problem," *Pattern Recognition*, vol. 34, no. 2, pp. 415-424, 2001.
- [6] D. W. Van der Merwe and A. P. Engelbrecht, "Data clustering using particle swarm optimization," in *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'03)*, Canberra, Australia, 2003, pp. 215-220.
- [7] P. S. Shelokar, V. K. Jayaraman, and B. D. Kulkarni, "An ant colony approach for clustering," *Analytica Chimica Acta*, vol. 509, no. 2, pp. 187-195, 2004.
- [8] S. C. Satapathy and A. Naik, "Data clustering based on teaching-learning-based optimization," in *Swarm, Evolutionary, and Memetic Computing*. Heidelberg: Springer, 2011, pp. 148-156.
- [9] Y. Kumar and G. Sahoo, "A charged system search approach for data clustering," *Progress in Artificial Intelligence*, vol. 2, no. 2-3, pp. 153-166, 2014.
- [10] Y. Kumar and G. Sahoo, "A chaotic charged system search approach for data clustering," *Informatica*, vol. 38, no. 3, pp. 249-261, 2014.
- [11] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm," *Applied Soft Computing*, vol. 11, no. 1, pp. 652-657, 2011.
- [12] K. S. Al-Sultan, "A tabu search approach to the clustering problem," *Pattern Recognition*, vol. 28, no. 9, pp. 1443-1451, 1995.
- [13] K. Krishna and M. N. Murty, "Genetic K-means algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 29, no. 3, pp. 433-439, 1999.

- [14] Y. T. Kao, E. Zahara, and I. W. Kao, "A hybridized approach to data clustering," *Expert Systems with Applications*, vol. 34, no. 3, pp. 1754-1762, 2008.
- [15] S. Rana, S. Jasola, and R. Kumar, "A review on particle swarm optimization algorithms and their applications to data clustering," *Artificial Intelligence Review*, vol. 35, no. 3, pp. 211-222, 2011.
- [16] C. Y. Tsai and I. W. Kao, "Particle swarm optimization with selective particle regeneration for data clustering," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6565-6576, 2011.
- [17] C. Zhang, D. Ouyang, and J. Ning, "An artificial bee colony approach for clustering," *Expert Systems with Applications*, vol. 37, no. 7, pp. 4761-4767, 2010.
- [18] X. Yan, Y. Zhu, W. Zou, and L. Wang, "A new approach for data clustering using hybrid artificial bee colony algorithm," *Neurocomputing*, vol. 97, pp. 241-250, 2012.
- [19] M. Laszlo and S. Mukherjee, "A genetic algorithm that exchanges neighboring centers for k -means clustering," *Pattern Recognition Letters*, vol. 28, no. 16, pp. 2359-2366, 2007.
- [20] Y. Hong and S. Kwong, "To combine steady-state genetic algorithm and ensemble learning for data clustering," *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1416-1423, 2008.
- [21] A. J. Sahoo and Y. Kumar, "Modified teacher learning based optimization method for data clustering," in *Advances in Signal Processing and Intelligent Recognition Systems*. Heidelberg: Springer, 2014, pp. 429-437.
- [22] T. Niknam and B. Amiri, "An efficient hybrid approach based on PSO, ACO and k -means for cluster analysis," *Applied Soft Computing*, vol. 10, no. 1, pp. 183-197, 2010.
- [23] S. C. Chu, P. W. Tsai, and J. S. Pan, "Cat swarm optimization," in *PRICAI 2006: Trends in Artificial Intelligence*. Heidelberg: Springer, 2006, pp. 854-858.
- [24] B. Santosa and M. K. Ningrum, "Cat swarm optimization for clustering," in *Proceedings of IEEE International Conference of Soft Computing and Pattern Recognition (SOCPAR'09)*, Malacca, Malaysia, 2009, pp. 54-59.
- [25] G. Panda, P. M. Pradhan, and B. Majhi, "IIR system identification using cat swarm optimization," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12671-12683, 2011.
- [26] P. M. Pradhan and G. Panda, "Solving multiobjective problems using cat swarm optimization," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2956-2964, 2012.
- [27] Y. Kumar and G. Sahoo, "A hybridize approach for data clustering based on cat swarm optimization," *International Journal of Information and Communication Technology*, 2014 (accepted).
- [28] P. W. Tsai, J. S. Pan, S. M. Chen, and B. Y. Liao, "Enhanced parallel cat swarm optimization based on the Taguchi method," *Expert Systems with Applications*, vol. 39, no. 7, pp. 6309-6319, 2012.
- [29] X. Hu, R. C. Eberhart, and Y. Shi, "Swarm intelligence for permutation optimization: a case study on n -queens problem," in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium (SIS'03)*, Indianapolis, IN, 2003, pp. 243-246.
- [30] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82-102, 1999.
- [31] S. C. Chu and P. W. Tsai, "Computational intelligence based on behaviors of cats," *International Journal of Innovative Computing, Information and Control*, vol. 3, no. 1, pp. 163-173, 2007.
- [32] H. Wang, Y. Liu, C. Li, and S. Zeng, "A hybrid particle swarm algorithm with Cauchy mutation," in *Proceedings of IEEE Swarm Intelligence Symposium (SIS2007)*, Honolulu, HI, 2007, pp. 356-360.
- [33] S. Rahnamayan, H. R. Tizhoosh, and M. M. Salama, "Opposition-based differential evolution algorithms," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC2006)*, Vancouver, Canada, 2006, pp. 2010-2017.
- [34] S. Rahnamayan, H. R. Tizhoosh, and M. M. Salama, "Opposition-based differential evolution," *IEEE Transaction on Evolutionary Computation*, vol. 12, no. 1, pp. 64-79, 2008.
- [35] H. Wang, H. Li, Y. Liu, C. Li, and S. Zeng, "Opposition-based particle swarm algorithm with Cauchy mutation," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC2007)*, Singapore, 2007, pp. 4750-4756.

- [36] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, and M. Ventresca, "Enhancing particle swarm optimization using generalized opposition-based learning," *Information Sciences*, vol. 181, no. 20, pp. 4699-4714, 2011.



Yugal Kumar

He received his B.Tech. in Information Technology from Maharishi Dayanand University, Rohtak (India) in 2006 & M.Tech. in Computer Engineering from same in 2009. At present, he is pursuing Ph.D. in Department of Computer science and engineering at Birla Institute of Technology, Mesra, Ranchi, India. His research interests include fuzzy logic, computer network, Data Mining and Swarm Intelligence.



Gadadhar Sahoo

He received his M.Sc. in Mathematics from Utkal University in the year 1980 and Ph.D. in the Area of Computational Mathematics from Indian Institute of Technology, Kharagpur in 1987. He has been associated with Birla Institute of Technology, Mesra, Ranchi, India since 1988, and currently, he is working as a Professor and Head in the Department of Computer Science. His research interest includes theoretical computer science, parallel and distributed computing, cloud computing, evolutionary computing, information security, image processing and pattern recognition.