Evolutionary Intelligence (2020) 13:593–609
https://doi.org/10.1007/s12065-020-00373-0

RESEARCH PAPER

A neighborhood search based cat swarm optimization algorithm for clustering problems

Hakam Singh[1] · **Yugal Kumar**[1]

Received: 17 May 2019 / Revised: 17 January 2020 / Accepted: 20 February 2020 / Published online: 27 February 2020

Abstract
Clustering is an unsupervised technique that groups the similar data objects into a single subset using a distance function. It is also used to find the optimal set of clusters in a given dataset and each cluster consists of homogenous data objects. In present work, an algorithm based on cat swarm optimization (CSO) is adopted for finding the optimal set of cluster centers for allocating the data objects. Further, some improvements are also incorporated in CSO algorithm for improving clustering performance. These modifications are described as an improved solution search equation to improve convergence rate and an accelerated velocity equation for balancing exploration and exploitation processes of CSO algorithm. Moreover, a neighborhood-based search strategy is introduced to handle local optima problem. The performance of proposed algorithm is tested on eight real-life datasets and compared with well-known clustering algorithms. The simulation results showed that proposed algorithm provides quality results in comparison to existing clustering algorithms.

Keywords Cat swarm optimization · Clustering · Machine learning · Meta-heuristics

Abbreviations

ABC	Artificial bee colony
ACO	Ant colony optimization
BATC	Bat algorithm based clustering
CABC	Cooperative artificial bee colony
CDC	Count to dimensions
CPSO	Cooperative particle swarm optimization
CS	Cuckoo search
CSO	Cat swarm optimization
DE	Differential evolution
FPAC	Flower pollination algorithm based clustering
GA	Genetic algorithm
GQCS	Genetic quantum cuckoo search
GWA	Grey wolf algorithm
HABC	Hybrid artificial bee colony
HCSDE	Hybrid cuckoo search and differential evolution
HS	Harmony search
KCPSO	K-means chaotic particle swarm optimization
KFCM	Kernel based fuzzy C-means
KHM	K-harmonic means
KICS	K-means and improved cuckoo search
MO	Magnetic optimization
M-TLBO	Modified-teaching learning based optimization
PSO	Particle swarm optimization
QPSO	Quantum-behaved particle swarm optimization
R	Rejected
SA	Simulated annealing
SMP	Seeking memory pool
SRD	Seeking range of selected dimension
TLBO	Teaching learning based optimization
TS	Tabu search

✉ Yugal Kumar
yugalkumar.14@gmail.com

Hakam Singh
hakamsingh011@gmail.com

[1] Department of Computer Science and Engineering, JUIT, Waknaghat, Himachal Pradesh, India

1 Introduction

In the field of machine learning, clustering is a popular unsupervised data analysis technique. In clustering, data objects are grouped into different subsets in optimal manner [1–3]. Further, clustering is classified as partitional clustering, hierarchical clustering, grid-based clustering, density-based clustering and model-based clustering [4, 5]. The partitional clustering divides the dataset into several disjoint groups that are optimal in terms of some predefined criteria. The objective of partitional clustering is to maximize the intra-cluster compactness and minimizing inter-cluster likeness. In hierarchical clustering, clustered are formed in terms of tree structure and each node represent a cluster. Furthermore, two approaches, agglomerative (bottom up) and divisive (top down) are mentioned in hierarchal clustering. In agglomerative approach, initially, each data object is taken as separate cluster and data objects merge with other data objects using a similarity function. While in divisive approach, all data points belong to one cluster and divide into smaller clusters using some dissimilarity function in successive iterations. In grid-based clustering, data space divides into finite number of cells to form a grid like structure. Density-based clustering describes using the compactness of data objects. The density can be defined as number of data objects present in a given radius of an object. While, model-based clustering considers the probability distributions to generate clusters. Clustering techniques have been widely adopted in diverse field such as bio-engineering, stock market, pattern recognition, image processing and medical data analysis [6–8]. In past few decades, large number of clustering algorithms have been reported for cluster analysis such as particle swarm optimization algorithm [9], magnetic optimization algorithm [10], charged system search algorithm [11], black hole algorithm [12], artificial bee colony algorithm [13, 14], ant colony optimization [15], big bang-big crunch algorithm [16] etc. These techniques generate efficient and effective in solution for clustering problems, but several shortcomings are associated with these techniques like stuck in local optima, slow convergence speed and diversity problems. Several researchers have addressed these problems and developed the new variants for cluster analysis. Some of these are summarized as Yang et al. [17] hybridized the particle swarm optimization algorithm through K-harmonic means to accelerate its convergence speed. Furthermore, Chuang et al. [18] incorporated chaotic maps in particle swarm optimization algorithm for improving convergence speed. The exploration capability of magnetic charge system search algorithm is enhanced through particle swarm optimization [19]. Xue et al. [20] developed a self-adaptive mechanism to handle local optima and poor exploration issues of artificial bee colony algorithm. Huang et al. [21] integrated ant colony optimization algorithm with particle swarm optimization to enhance its search ability. Jordehi [22] introduced chaotic maps in big bang-big crunch algorithm to handle local optima problem. Apart from these, ensembles methods are reported in literature [23].

Recently, CSO algorithm is gain wide popularity among research community. This algorithm has been applied to solve variety of optimization problems such as numerical problems optimization [24], linear and circular antenna arrays synthesis [25, 26], neural network optimization [27], workflow scheduling in cloud [28], wireless sensor networks [29], solar photovoltaic cells [30], image analysis [31, 32] etc. It is observed that CSO algorithm has stronger exploration capability as compared same class of algorithms [33]. The motivation to choose the CSO algorithm for cluster analysis is its strong exploration capability. But, this algorithm also suffers with slow convergence rate and traps in local optima sometimes [34, 35]. Furthermore, it is noticed that CSO algorithm have weak exploitation capability [33]. Several issues that can affect the performance of CSO algorithm are summarized as

- Slow convergence speed due to absentia of global best position of cat in search space
- Lack of coordination between exploration and exploitation processes
- Sometime suffered with weak diversification and local optima due to lack of information exchange mechanism

The main contribution of this work is to address the aforementioned issues and also improve the performance of CSO algorithm especially for cluster analysis. Some improvements are incorporated in CSO to make it more robust, viable and efficient for cluster analysis. These improvements are given as

- A new position update equation is proposed using the concept of global best position of cats to enhance the convergence speed.
- A new accelerated velocity equation is designed using the concept of Levy flight for balancing the coordination between exploration and exploitation processes.
- A neighborhood search strategy is introduced to explore optimum solution and also to handle local optima problem.

The proposed algorithm is taken into consideration for solving real world clustering problems. The main objective of proposed improvements is to obtain optimum clustering results with minimized intra cluster distance.

2 Related works

In past few decades, large number of evolutionary algorithms have been designed to solve clustering problems. This section presents the recent works reported on clustering problems. Zhang et al. [13] developed a metaheuristic algorithm inspired through bee's behavior for solving clustering problems and also categorized the bees into three classes such as employed bees, onlooker bees and scout bees. Among these, first two classes are accountable for global search, while scout bee class is accountable for local search. The performance of this algorithm is tested over three benchmark datasets and compared with TS, GA, SA, K–NM–PSO and ACO algorithms. The experimental results showed that proposed algorithm effectively solves the clustering problems. Yan et al. [36] presented a hybrid version of artificial bee colony algorithm (HABC) for cluster analysis. The objective of HABC algorithm is to enhance the information exchange mechanism between bees. To achieve the same, crossover operator of GA is inherited in ABC algorithm. The performance of HABC algorithm is assessed using iris, wine, contraceptive method choice, wisconsin breast cancer, glass, liver disorder datasets and compared with ABC, CABC, PSO, CPSO, GA and K-means clustering algorithms. It is seen that HABC achieves best intra cluster distance value for five datasets except wisconsin breast cancer dataset. Cura [9] introduced particle swarm optimization algorithm in clustering field for obtaining optimal clustering results. PSO algorithm is effective in both conditions weather the number of clusters are known or unknown. The performance of PSO algorithm is tested over seven datasets and compared with K-NM-PSO, ACO and ABC algorithms. Author claimed that PSO algorithm performs well in clustering field. In order to design a robust algorithm for clustering task, Yang et al. [17] hybridized PSO algorithm with K-harmonic means, called PSOKHM. The aim of hybridized algorithm is to integrate the advantages of KHM i.e. better convergence rate and PSO algorithm i.e. efficient optimization for solving clustering problems. The performance of PSOKHM algorithm is tested over seven datasets and compared with PSO and KHM. It is noticed that PSOKHM is effective and efficient method to solve clustering problems. Siddiqi et al. [37] have introduced a new heuristic approach in clustering field. The proposed approach works in two modes. Initially, a greedy method is used to select initial seed points. Whereas, in second mode, a heuristic approach inspired from GA and SimE algorithm is designed for optimization task. The efficiency of this algorithm is tested over sixteen real-life numeric datasets and compared with GenSA, GA, and DE algorithms. The simulation results showed that the proposed meta-heuristic algorithm works efficiently in clustering field. A meta-heuristic algorithm inspired from class room teaching is developed for clustering problems [38]. This algorithm works in two phases i.e. teacher and learner phase. In teaching phase, student learns from teacher to improve his knowledge, while in learner phase, students interact with other learners. The chaotic maps are also combined in this algorithm to diversify the solution space. The efficiency of this algorithm is tested on five benchmark datasets and compared with K-Means, PSO, ACO, CSO, TLBO and M-TLBO clustering algorithms. It is observed that this algorithm provides state of art clustering results as compared other algorithms. Kumar et al. [39] considered grey wolf algorithm in clustering field to obtain optimal number of cluster sets. This algorithm is based on the leadership quality and hunting mechanism of wolfs. The performance of this algorithm is tested on eight datasets and compared with KM, GAC, HSC, MHSC, PSOC, FPAC and BATC algorithms. It is noticed that proposed algorithm achieves higher clustering results than others. Boushaki et al. [40] developed an extended version of cuckoo search algorithm to solve clustering problems. The quantum theory-based search mechanism is used to enhance global search capability of traditional cuckoo search algorithm (CSA). Moreover, chaotic maps are incorporated to diversify the solution space in place of random values. The performance of CSA is tested on six well-known datasets and compared with GA, QPSO, CS, GQCS, DE, KCPSO, HCSDE and KICS clustering algorithms. Authors claimed that proposed algorithm gives more accurate results for clustering problems. A new meta-heuristic clustering algorithm based on magnetic theory is reported in [10]. The performance of proposed MO algorithm is tested over twelve datasets and compared with HYBRID_DE, MIN_MAX, PSO, K-means and KFCM algorithms. Simulation results stated that MO algorithm attains more accurate results in comparison to other clustering algorithms. Kumar and Sahoo [41] presented a hybrid version of cat swarm optimization algorithm for clustering task. To enhance population diversity of traditional algorithm, a Monte Carlo based search equation is introduced. Moreover, a population centroid operator is added to handle local optima situation. The efficiency of this algorithm is tested on both artificial and real-life datasets and compared with GA, ACO, CSO, ICSO, PSO, and K-means clustering algorithms. Authors claimed that proposed algorithm determines better quality results than other clustering algorithms. To solve clustering problems Jadhav and Gomathi [42] integrated grey wolf optimizer with whale optimization algorithm. In this work, a new fitness function is designed to measure the correctness of candidate solution. The estimation procedure of fitness function is subjected on three performance measure i.e. inter-cluster distance, intra-cluster distance and cluster density. The candidate solution with minimal fitness value is considered as centroid. The performance of proposed algorithm is tested over three datasets and compared with well-known clustering algorithms. It is seen that proposed algorithm is an effective and efficient for solving clustering problems. In order to eliminate the inadequacies of K-means algorithm, Kumar et al. [43] integrated K-means algorithm with artificial bee colony algorithm. The work of ABC algorithm is to determine optimal cluster center for K-means algorithm. The performance of proposed algorithm is evaluated on six datasets and compared with two clustering algorithms. Authors claimed that proposed algorithm works efficiently and determines optimal cluster centers. A new clustering algorithm is reported to detect number of clusters in automatic manner [44]. In this algorithm, some heuristic rules are designed based on *k*-nearest neighbor's chain to discover number of clusters. The performance this algorithm is compared with both of real-life and synthetic datasets. It is observed that proposed algorithm is able to find exact number of clusters. To determine cluster number automatically, a clustering algorithm named as harmonious genetic clustering algorithm is reported in [45]. In this algorithm, harmonious mating is determined to select suitable mate for each chromosome. Moreover, three mating prohibition schemes is also developed to avoid illegal mating. The performance of this algorithm is tested on both artificial and real-life datasets and compared with four recent clustering methods. Authors claimed that proposed method automatically detects number of clusters with better quality. He and Tan [46] presented a two-stage genetic algorithm for automatic clustering. In this algorithm, the selection and mutation operators of genetic algorithm are used to search number of clusters with appropriate partitioning. Moreover, maximum attribute range partition approach is also adopted for initial population selection. The performance of the proposed algorithm is tested on both artificial and real-life datasets and compared with six well-known clustering algorithms. It is noticed that proposed algorithm works efficiently even without prior knowledge of cluster numbers.

3 Cat swarm optimization

Chu et al. developed a metaheuristic algorithm based on the behavior of cats for solving the complex optimization problems [33]. This algorithm works in two modes i.e. seeking and tracing. Seeking mode considers the resting behavior of cats, whereas, tracing mode reflects the haunting skills of cats. The working of CSO algorithms are discussed through aforementioned modes.

3.1 Seeking mode

The seeking describes the resting behavior of cats, but still cats are attentive towards its target. The resting behavior of cats represents through the position vector and furthermore, cats change its positions in slow manner. This mode can be acted as local search and entire search space is explored in slowly fashion to determine optimum solutions. Several notations are described in this mode which are given as

- Seeking Memory Pool (SMP): It denotes the number of replicated copies (position movement) of the cat.
- Seeking Range of selected Dimension (SRD): It denotes the difference between old and new positions of cat selected for mutation.
- Counts of Dimension to Change (CDC): It denotes the number of positions of a cat undergone for mutation.

Steps involved in seeking modes are listed as:

1. Define the number of copies (T) of the i^{th} cat to be replicated.
2. Initialize the CDC constraint, do following
    i. Add/Subtract SRD value of current position of cat in random order.
    ii. Replace old values for all copies.
3. Compute the fitness function for all replicated copies of cat.
4. Select the best candidate solution and deploy at the position of i^{th} cat.

3.2 Tracing mode

The tracing mode of algorithm describes the hunting skills of cat. When, a cat hunts the prey, the position of cat changes due to movement and velocity vector of cat is also updated. The updated velocity of cat is computed using Eq. 1.

$$V^d_{j\,new} = w*V^d_j + c*r*\left(X^d_{jbest} - X^d_j\right) \quad (1)$$

Where $V^d_{j\,new}$, V^d_j represents the new and old velocity values of j^{th} cat in the d^{th} dimension, w is a weight factor between 0 and 1, r is a random value, c is user defined parameter, X^d_{jbest} denotes the best position attained by the j^{th} cat and X^d_j represent the current position of the j^{th} cat and $d = 1, 2, \ldots D$. The updated position of cat is computed using Eq. 2.

$$X^d_{j\,new} = X^d_j + V^d_j \quad (2)$$

Where $X^d_{j\,new}$ represents the newly updated position j^{th} cat, X^d_j represent the current position and V^d_j denotes the old velocity values of j^{th} cat in the d^{th} dimension.

4 Proposed improvements in CSO algorithm

This section discusses the improvements proposed in traditional CSO algorithm. To improve the performance of CSO algorithm and make it more robust and efficient for cluster analysis, several improvements are taken into consideration. Section 4.1 discusses the proposed improvements in CSO algorithm. Whereas, in Sect. 4.2, a neighborhood search strategy procedure is discussed to generate optimum candidate solution with respect to current cat position as well as address the local optima problem. The working steps of proposed CSO algorithm are presented in Sect. 4.3.

4.1 Proposed improvements

In CSO algorithm, the positions of cats are updated using current positions and velocities of cats [47]. It is noticed that sometime algorithm cannot explore entire search space for optimum solution due to lack of global best cat position information [48]. In turn, convergence rate of CSO algorithm is affected. It is reported that CSO algorithm having good exploration capability, but suffers with weak exploitation ability [24, 49]. But, to obtain optimum results, both of abilities should be well balanced. The following improvements are incorporated in CSO algorithm to obtain quality clustering results.

- To enhance the convergence rate of CSO algorithm, a modified search equation is proposed to guide the position of cats towards the global best or optimum solution. So, the global best position and levy flight components are integrated in position search equation of CSO algorithm. Hence, a new position update equation based on global best information and levy flight is presented in Eq. 3.

$$X_{j\,new}^{d+1} = X_j^d + (P_g * rand()) \oplus Levy \tag{3}$$

Levy (s) $\sim |s|^{-1-\beta}$ where β is an index value having range ($0 < \beta \leq 2$), Pg is global best position of cat and s is a step size $s = \frac{u}{|v|^{\frac{1}{\beta}}}$ The values of u, v are computed using Eqs. 4–6.

$$u \sim N\left(0, \sigma_u^2\right), v \sim N\left(0, \sigma_v^2\right) \tag{4}$$

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\beta\Gamma\left[(1+\beta)/2\right]2^{(\beta-1)/2}} \right\}^{1/\beta}, \quad \sigma_v = 1 \tag{5}$$

Where $\Gamma(1+\beta) = \int_0^\infty t^\beta e^{-t} dt$.

- To balance the coordination between exploration and exploitation abilities, a new accelerated velocity equation is designed, especially for tracing mode of CSO algorithm. In proposed CSO algorithm, velocity of cats is computed using Eq. 6.

$$V_{j\,new}^{d+1} = V_j^d + \left(P_{best,j} - X_j^d\right) rand() \oplus Levy \tag{6}$$

Where $V_{j\,new}^{d+1}$, V_j^d represents the new and old velocity values, X_j^d represent the current position and $P_{best,j}$ is the personal best position of j^{th} cat.

4.2 Neighborhood search strategy

The neighborhood structures can improve the efficiency and search quality of algorithms. The concept for neighborhood search has been successfully adopted for improving the efficiency of differential evolution (DE), firefly algorithm (FA) PSO and ABC [50–53]. This method increases the probability of finding efficient solutions and also enables to handle local optima situation. To enhance the performance of traditional CSO algorithm and generate diversify population to overcome local optima problem, three neighborhood search criteria are incorporated into proposed CSO algorithm. The neighborhood search strategy includes both of local and global searches strategies. In this work, one local search and two global searches are performed to find feasible candidate solutions. Let us consider that there are N numbers of cats organized in a circle according their indices and X_3 and X_1 are two immediate neighbors of X_2. It is assumed that the population of cats is ten. Based on circle representation, a concept of k-neighborhood is mentioned in Fig. 1a, b where k-neighborhood (k = 2) and number of cats are nine in the neighborhood of X_i. For each cat X_i, its k-neighborhood consists of 2 k +1 cats $X_{i-k}, \ldots, X_i, \ldots, X_{i+k}$, where k is an integer $0 \leq k \leq N - 1/2$.

Hence, to discover more feasible candidate solution, neighborhood-based search technique is applied in tracing mode of CSO algorithm. For each cat, its neighborhood may have better candidate solutions. To enhance exploitation search ability, a local neighborhood search can be defined as follows.

$$X_i^1 = r_1 \times X_i + r_2 \times best_{cat_i} + r_3 \times \left(X_{i1} - X_{i2}\right) \tag{7}$$

Where X_{i1} and X_{i2} are two cats randomly nominated from the k-neighborhood radius of Xi ($i1 \neq i2 \neq i$), $best_{cat_i}$ is the best position of cat, r_1, r_2 and r_3 are random values between (0,1) and have total $r_1 + r_2 + r_3 = 1$.

Further, a global neighborhood search, similar to local neighborhood search is developed to improve exploration search ability specified in Eq. 8.

$$X_i^2 = r_4 \times X_i + r_5 \times best_{cat_i} + r_6 \times \left(X_{i3} - X_{i4}\right) \tag{8}$$

Where X_{i3} and X_{i4} are two cats randomly nominated from the entire population ($i3 \neq i4 \neq i$), $best_{cat_i}$ is the global best position of cat, r_4, r_5 and r_6 are random values between (0,1) and have total $r_4 + r_5 + r_6 = 1$. To solve local optima problem, a Cauchy mutation operator is adopted in the proposed approach [54]. The long tail Cauchy distribution (random variables) enables the algorithm to come out from local optima situation.

$$X_i^3 = X_i + Cauchy() \tag{9}$$

Where, Cauchy () is a random value taken from Cauchy distribution. In neighborhood identification, three trial solutions X_i^1, X_i^2 and X_i^3 are generated using Eqs. 7–9. Then, the best solution among X_i^1, X_i^2 and X_i^3 is selected as the new X_i. Figure 2a–c shows the exploration of search space to obtain

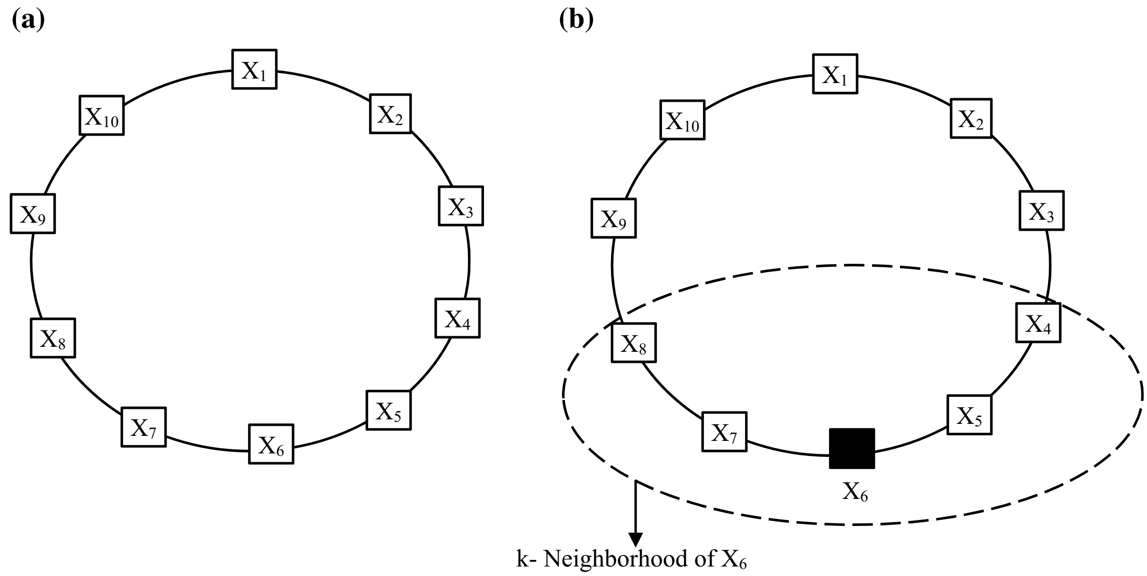


Fig. 1 **a** Circle representation of each cat having two neighboring cats, **b** Represents k-neighborhood structure, where number of cats = 10 and k = 2

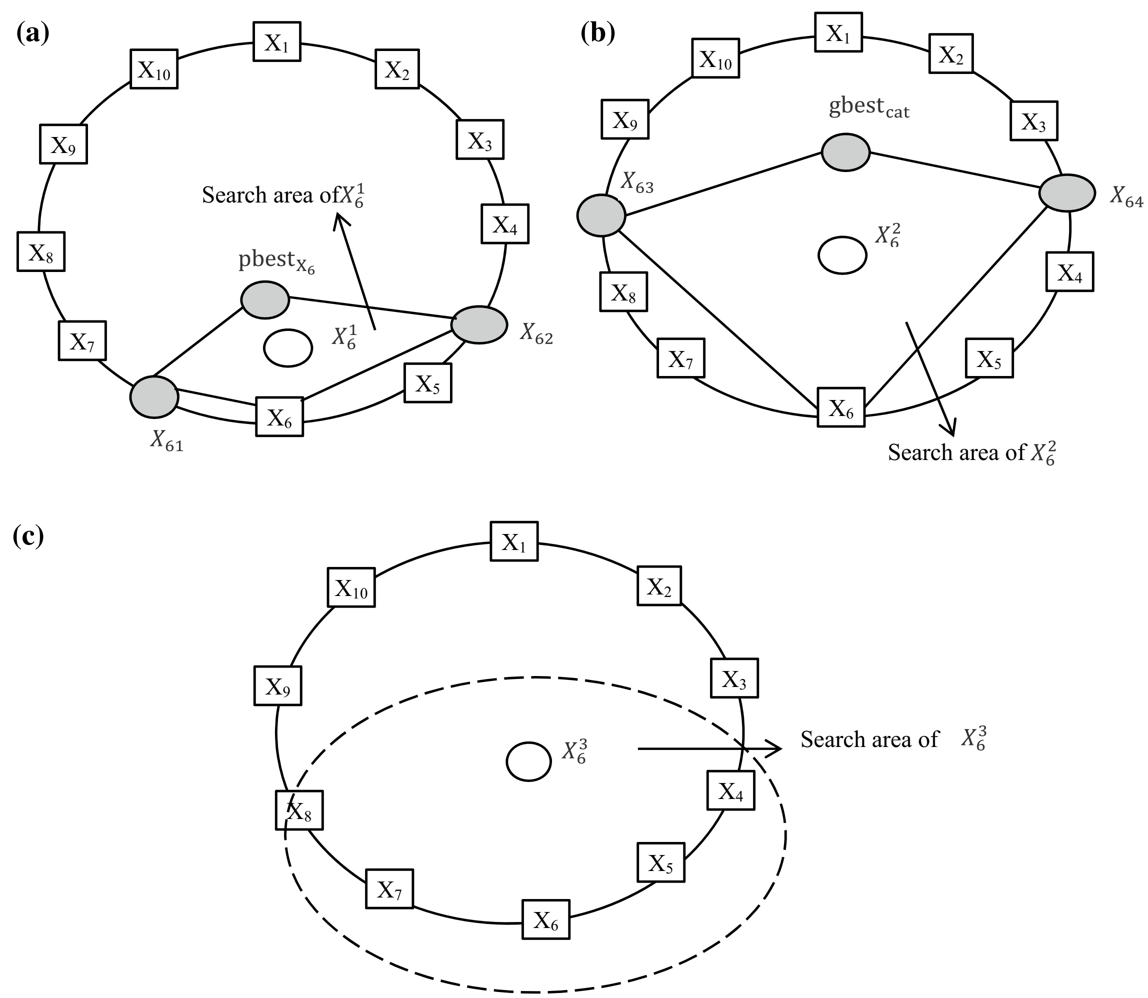


Fig. 2 Searching mechanism of local candidate solution, global candidate solution and a feasible candidate solution to overcome local optima

good candidate solution during the different aspect of the proposed algorithm.

The steps of neighborhood search are given below.

4.3 Steps of enhanced CSO algorithm for clustering

This subsection describes the detailed steps of proposed cat swarm optimization algorithm for clustering problems.

Algorithm 1: Neighborhood Search Strategy

Begin

Step 1: Initialize the initial solution X_i, K- Neighborhood structures, function f to evaluate solutions.

Step 2: Generate the three trial candidate solutions X_i^1, X_i^2 and X_i^3 Using Eqs. 7-9.

Step 3: Calculate the fitness values of new generated candidate solutions X_i^1, X_i^2 and X_i^3.

Step 4: FE=FE+3 /* no. of Fitness Evaluation (FE).

Step 5: Pick the best solution among X_i, X_i^1, X_i^2 and X_i^3 according to fitness values as new X_i.

End

Algorithm 2: Proposed Enhanced CSO Algorithm for Clustering

Step 1: Uniformly distribute N number of cats in search space and initialize different parameters viz population of cats, neighborhood structure, β, α, C, SMP and SRD.

Step 2: Initialize the position and velocity of each cat into the D-dimensional solution search space.

Step 3: Calculate the fitness value of cats and place the best position of cat into memory.

Step 4: While (i < maximum number of iterations /* i = iteration number), do

Step 5: In accordance with Flag value, randomly distribute cats in search space.

Step 6: If (Flag==1) /* cat in seeking mode

- Apply seeking mode process.
- Construct the j^{th} copy of each cat.
- Using SRD, compute shifting bit value for each cat.
- Add/Subtract SRD value of current position of cat in random order.
- Calculate the fitness function for each new position of cats.
- Compare the fitness function values and place the best position of cat into memory
- End

Step 7: Else /* cat in tracing mode

- Apply tracing mode process.
- Update the velocity of each cat using Eq. 6.
- Update the position of each cat using Eq. 3.
- Calculate the fitness function for each new position of cats.
- Compare the fitness function values and place the best position of cat into memory
- End

Step 8: Updates the position of cats and also determine the best position of cat.

Step 9: If $(rand(0,1) \leq Fit_i)$, then

Step 10: Apply the neighborhood search strategy (Algorithm 1).

Step 11: Updates the position of cats and (P_g) global best position.

Step 12: End

Step 13: $i = i + 1$

Step 14: End

Step 15: Obtain the final solution.

5 Result and discussion

This section illustrates the experimental results of the proposed CSO algorithm on real-world clustering problems. MATLAB 2016a environment is used to implement the proposed CSO algorithm for clustering problems. In order, to start this experiment, the various algorithmic parameters are initialized such as $\mathrm{SMP} = 10, \mathrm{MR} = 0.5, \mathrm{C} = 2, \beta \in 0.1{-}0.7, \alpha \in 0.1{-}0.5$, population size = 100 and value of ε is between 0 and 1. These constraints are kept constant during entire execution of program.

5.1 Performance evaluation and results

The performance of proposed algorithm is evaluated on eight real-life datasets. A detailed description of these datasets is given in Table 1. The simulation results of proposed CSO algorithm are compared with GA, PSO, ACO, CSO and K-means algorithms using intra cluster distance, f-measure and standard deviation.

Table 2 shows the experimental results of proposed CSO algorithm and other algorithms being compared. The results are taken over thirty independent runs and each run having dissimilar initial seed points. From simulation results, it is noticed that proposed CSO algorithm gives better average intra cluster distance for all datasets expect glass dataset. Furthermore, on the analysis of f-measure, it is seen that proposed CSO algorithm obtains larger f-measure values with most of datasets except glass and vowel datasets. Finally, it is stated that the proposed algorithm provides better-quality results than other clustering algorithms.

Figures 3a and 4a depict the iris dataset distribution view in two and three-dimensional space. While the Figs. 3b and 4b displays the clustering of data objects in different clusters with respect to the Figs. 3a and 4a using proposed CSO algorithm.

Figures 5a and 6a show the distribution of data objects in wine dataset. In Fig. 5a, two attributes of wine dataset such as malic acid and alcohol are used to show the distribution of data objects while, Fig. 6a considers the ash, malic acid and alcohol attributes. Figures 5b and 6b displays the clustering outcomes, where data items are grouped into three dissimilar clusters.

Figures 7a and 8a demonstrate the distribution of the data objects in CMC dataset using two and three attributes, while, the Figs. 7b and 8b show the clustered data objects of CMC dataset. Furthermore, to demonstrate the effectiveness of proposed algorithm with respect other algorithms, the conversion graphs are also plotted using intra cluster distance parameter. Figure 9a–h show the convergence behavior of proposed CSO algorithm and other algorithms being compared. Here, X-axis represents the no of iteration, whereas Y-axis represents the intra-cluster distance obtains in each iteration. It is observed that proposed algorithm provides better convergence rate except Glass and LD datasets. Hence, from these graphical representations it is clearly observed that proposed algorithm provides glowing convergence result among other clustering algorithms.

5.2 Statistical analysis

To validate the performance of proposed CSO, some statistical tests are also considered. The aim of these test is to identify substantial difference is existed between the performances of proposed algorithm and other clustering algorithm. To achieve the same, Friedman and Quade tests are adopted. Furthermore, a post hoc test (holm's test) is also conducted to confirm the significance of proposed algorithm. The level of confidence (α) is set as 0.1and0.05. Two hypotheses are designed to determine the significant difference between the performances of algorithms. These hypotheses are interpreted as hypothesis (H_0) and (H_1). Hypothesis (H_0) means algorithms are not different, hypothesis (H_1) means substantial difference exist.

Table 1 Detailed description of datasets

Dataset	Clusters (K)	Attributes (D)	Data instances (N)	Data instance in individual cluster
Iris	3	4	150	(50, 50, 50)
Wine	3	13	178	(59, 71, 48)
CMC	3	9	1473	(629,334, 510)
Cancer	2	9	683	(444, 239)
Glass	6	9	214	(70,17, 76, 13, 9, 29)
Thyroid	3	5	215	(150, 30, 35)
Vowel	6	3	871	(72, 89, 172, 151, 207, 180)
Wine	3	13	178	(59, 71, 48)

Table 2 Comparison of simulation results of proposed CSO and other clustering algorithms

Dataset	Parameters	Algorithms					
		K-means	GA	PSO	ACO	CSO	Proposed CSO
Iris	Best	97.12	113.98	96.48	96.89	96.94	96.08
	Avg.	112.44	125.19	98.56	98.28	97.86	97.18
	Worst	122.46	139.77	99.67	99.34	98.58	97.83
	SD	15.326	14.563	0.467	0.426	0.392	0.26
	F-measure	0.781	0.774	0.780	0.779	0.776	0.784
Cancer	Best	2989.46	2999.32	2978.68	2983.49	2985.16	2972.36
	Avg.	3248.25	3249.46	3116.64	3178.09	3124.15	3045.93
	Worst	3566.94	3427.43	3358.43	3292.41	3443.56	3282.75
	SD	256.58	229.734	107.14	93.45	128.46	56.24
	F-measure	0.832	0.819	0.826	0.829	0.831	0.833
CMC	Best	5828.25	5705.63	5792.48	5756.42	5712.78	5689.16
	Avg.	5903.82	5756.59	5846.63	5831.25	5804.52	5778.14
	Worst	5974.46	5812.64	5936.14	5929.36	5921.28	5914.25
	SD	49.62	50.369	48.86	44.34	43.29	39.54
	F-measure	0.337	0.324	0.333	0.332	0.334	0.336
Wine	Best	16,768.18	16,530.53	16,483.61	16,448.35	16,431.76	16,372.02
	Avg.	18,061.24	16,530.53	16,417.47	16,530.53	16,395.18	16,357.89
	Worst	18,764.49	16,530.53	16,594.26	16,616.36	16,589.54	16,556.76
	SD	796.13	0	88.27	48.86	62.41	41.78
	F-measure	0.519	0.515	0.516	0.522	0.521	0.523
Glass	Best	222.43	272.37	267.56	261.22	256.53	261.47
	Avg.	246.51	282.32	275.71	273.46	278.44	269.61
	Worst	258.38	291.77	284.52	293.08	282.27	274.24
	SD	18.32	4.138	8.59	6.58	15.43	9.24
	F-measure	0.426	0.333	0.412	0.402	0.416	0.424
Thyroid	Best	13,956.83	11,576.29	10,354.56	10,085.82	10,585.91	10,285.29
	Avg.	14,133.14	12,218.82	11,149.70	10,758.13	10,687.56	10,358.46
	Worst	14,642.21	13,254.39	13,172.86	12,134.82	11,934.34	11,564.02
	SD	246.06	32.64	27.13	21.34	98.62	72.34
	F-measure	0.731	0.763	0.778	0.781	0.774	0.783
Vowel	Best	152,422.26	152,234.73	151,976.01	149,395.60	152,436.58	148,826.95
	Avg.	159,642.89	159,353.49	157,999.82	158,458.14	158,956.81	157,328.93
	Worst	161,236.81	165,991.65	158,121.18	160,632.82	160,539.82	165,939.82
	SD	916	3105.54	2881.35	3485.38	3216.56	2691.23
	F-measure	0.652	0.643	0.645	0.648	0.646	0.651
LD	Best	11,397.83	532.48	209.15	224.76	231.54	214.76
	Avg.	11,673.12	543.69	236.47	241.23	240.16	232.58
	Worst	12,043.12	563.26	239.11	256.44	261.06	243.43
	SD	667.56	41.78	29.38	23.46	20.46	17.46
	F-measure	0.467	0.482	0.491	0.487	0.485	0.496

5.2.1 Using intra-cluster distance

This subsection describes the results of stastistical tests performed on average intra-cluster distance parameter. Table 3 demonstrates the average ranking of algorithms obtained through Friedman test, Table 4 presents the statistical outcomes obtained using Friedman and Quade test. The critical values for Friedman test are 11.070504 and 9.236361. The computed p values for Friedman test is 0.0005025. Hence, null hypothesis is strongly rejected and it is stated that

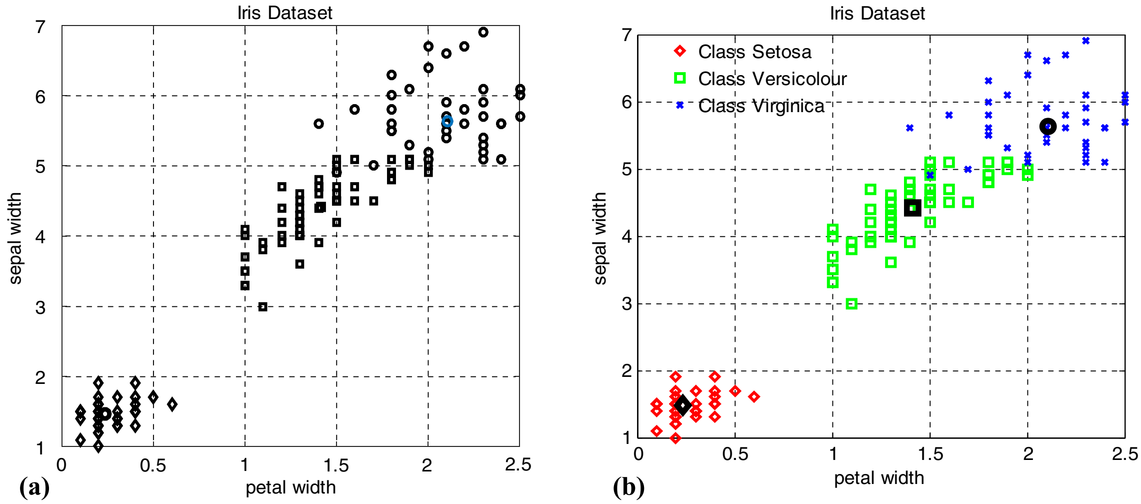


Fig. 3 **a** Iris dataset distribution view in 2D space. **b** Iris dataset clustering view in 2D space

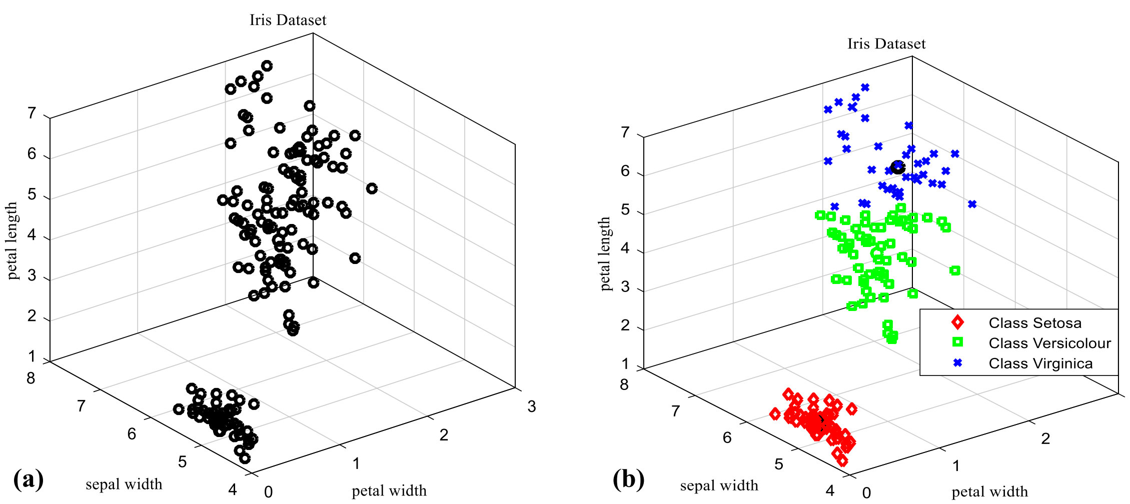


Fig. 4 **a** Iris dataset distribution view in 3D space. **b** Iris dataset clustering view in 3D space

substantial difference occurrs between the performances of proposed CSO and other compared algorithms.

Quade test is a progressive nonparametric test that provides excellent results in comprasion to parametric tests. The statistical results of Quade test are illustrated in Tables 4, 5 and 6. The statistical outcomes of Quade test are reported into Tables 4. While, a complete description about relative size and ranking of each algorithm is provided in Tables 5 and 6. The critical values for Quade test at confedence levels 0.05 and 0.1 are 2.485142 and 2.019125. The p value for Quade test is 7.42E−6 which rejects the null hypothesis and conclude that at least one of the algorithm is different from others.

In this work, Holm's method considers as a post hoc test and it is performed on both of statistical tests. The results of post hoc test are demonstrated in Table 7 and 8. It is noticed that hypothesis are rejected for all algorithms at confedence level 0.05 and 0.1. Hence, it is concluded that proposed CSO algorithm is statistically superior than rest of algorithms.

5.2.2 Using F-measure

This subsection describes the statistical results of Friedman and Quade tests using f-measure parameter. The results of these tests are illustrated in Tables 9, 10, 11 and

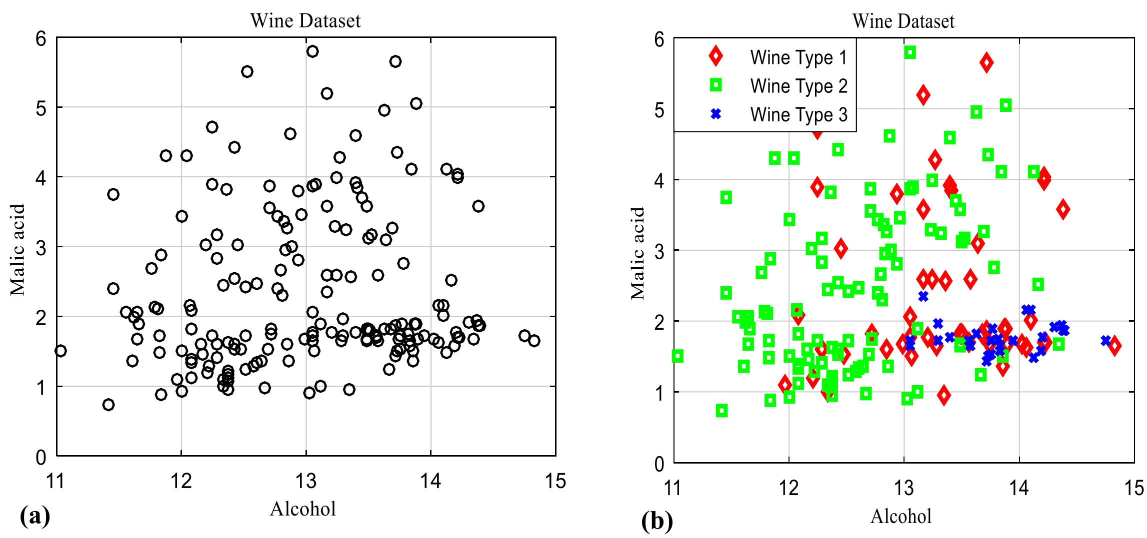


Fig. 5 **a** Wine dataset distribution view in 2D space. **b** Wine dataset clustering view in 2D space

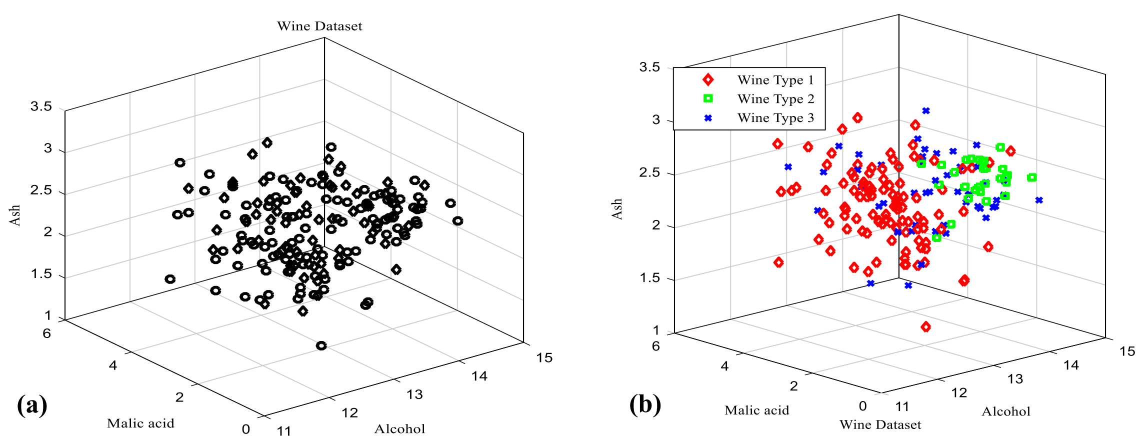


Fig. 6 **a** Wine dataset distribution view in 3D space. **b** Wine dataset clustering view in 3D space

12. The average ranking of all algorithms using Friedman test are reported in Table 9. It is noted that proposed algorithm obtains best ranking i.e. 1.13 among all other algorithms. Whereas, genetic algorithm having worst rank i.e. 5.5. The *p* value for Friedman test is 0.000321, which rejects the null hypothesis. The statistical outcomes of Quade test are stated into Tables 10, 11 and 12. Table 10 shows the statistics of the Quade test, whereas, Tables 12 and 13 illustrates the relative size of datasets and ranking of algorithms. The *p* value for Quade test is 7.921E−6 and it rejects the null hypothesis strongly. Finally, it is stated that the performance of proposed CSO algorithm is significantly differ from other algorithms.

Again, a post hoc test is performed to prove the substantial difference among the performance of proposed algorithm and other compared algorithms. The results of post hoc test are demonstrated in Tables 13 and 14. It is observed that null hypothesis is rejected at the confidence level 0.05 and 0.1. It is proved that the proposed algorithm is substantially different from rest of algorithms. Hence, proposed CSO algorithm is one of efficient and effective algorithm for solving the clustering problems.

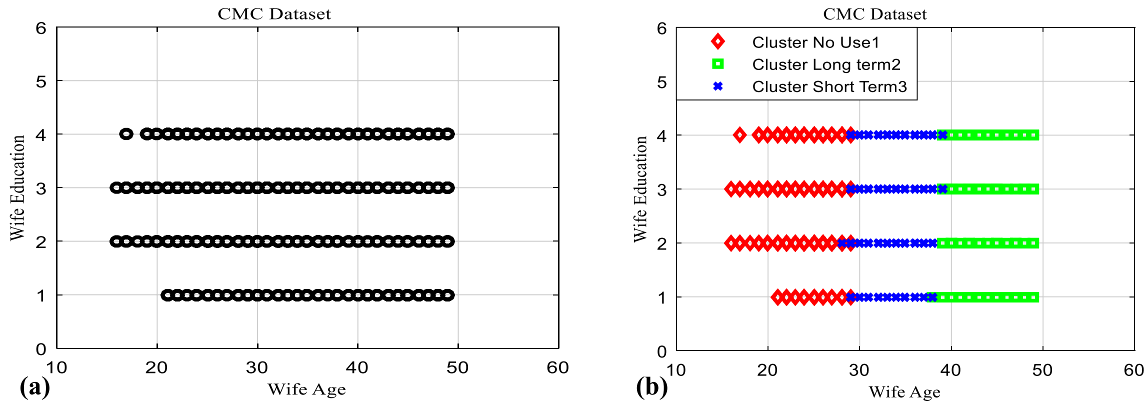


Fig. 7 **a** CMC dataset distribution view in 2D space. **b** CMC dataset clustering view in 2D space

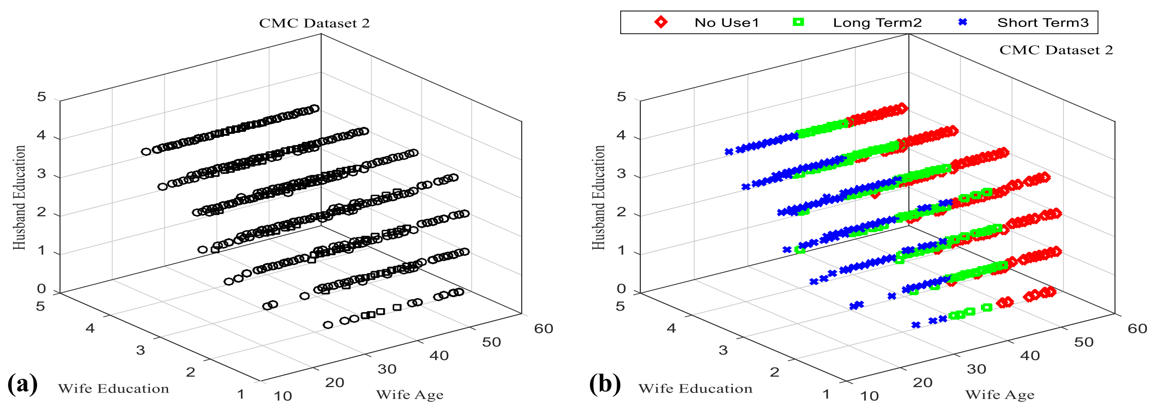
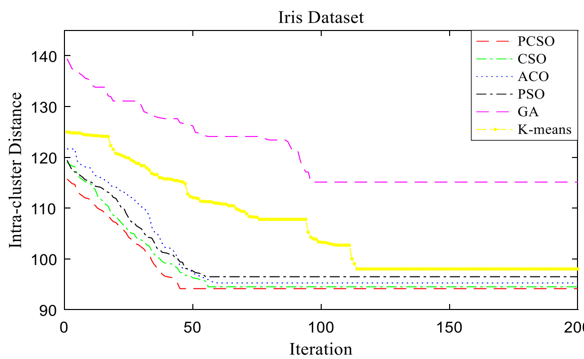


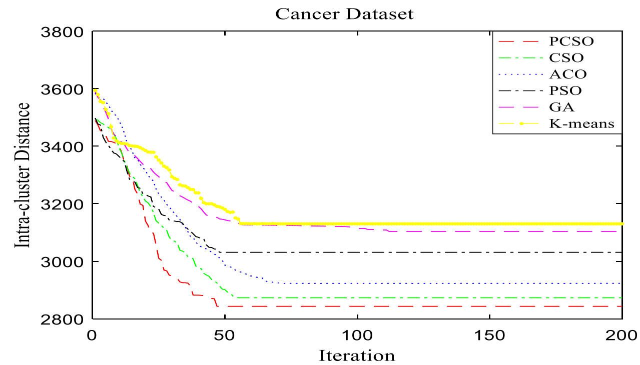
Fig. 8 **a** CMC dataset distribution view in 3D space. **b** CMC dataset clustering view in 3D space

6 Conclusion and future scope

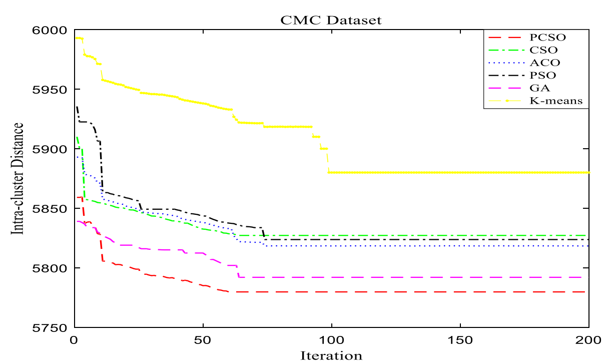
In this work, an enhanced variant of CSO algorithm is proposed to solve real world clustering problems. In order to make CSO algorithm more efficient and robust, several improvements are proposed. These improvements are described in terms of an accelerated velocity and improved solution search equations, especially tracing mode of CSO algorithm. The aim of these improvements is to make the coordination between exploration and exploitation processes and also improve the convergence rate of CSO algorithm. Further, a neighborhood-based search strategy is introduced to discover more efficient candidate solutions in search space. The performance of proposed CSO algorithm is examined over eight real-life datasets. From experimental results, it is clearly observed that the proposed CSO algorithm delivers good results in comparison to other algorithms. It is also observed that proposed algorithm is also capable to handle local optima situation. From experimentation section, it is concluded that proposed CSO algorithm achieves good quality results for clustering problems. Hence, it can be concluded that proposed algorithm can be widely used for data analysis. In future, the variable neighborhood strategy can be incorporated in CSO algorithm to determine optimal solutions. The different optimization strategies such as Monte Carlo, Hook and Jeev, linear programming and sequential search can be integrated with CSO algorithm to overcome its shortcomings. Furthermore, CSO algorithm can be used to solve real-life optimization problems such as workflow scheduling, classifications, path planning etc.



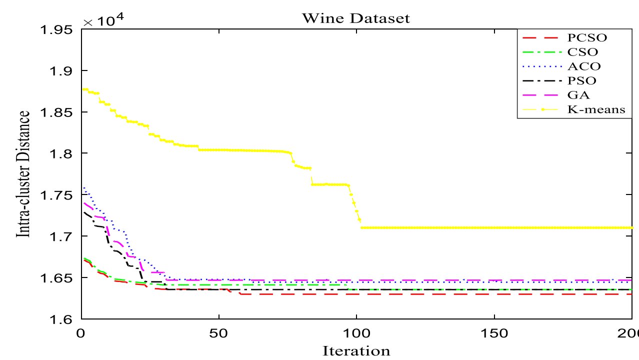
(a) Convergence behavior of all algorithms using iris dataset



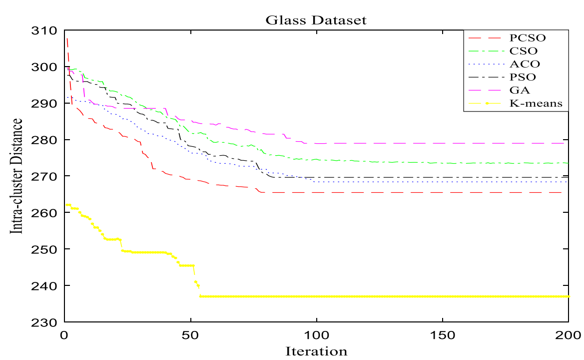
(b) Convergence behavior of all algorithms cancer iris dataset



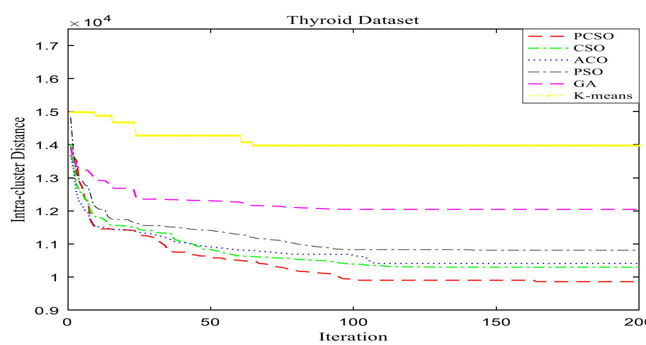
(c) Convergence behavior of all algorithms using CMC dataset



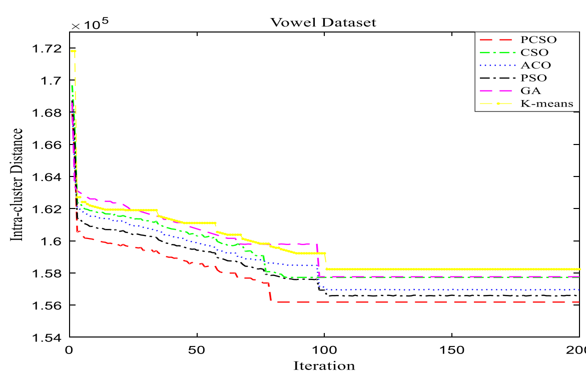
(d) Convergence behavior of all algorithms using wine dataset



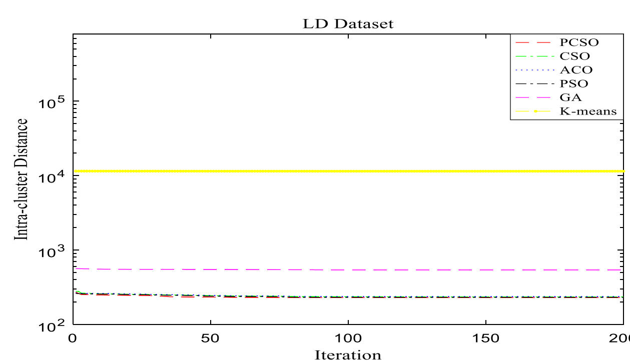
(e) Convergence behavior of all algorithms using glass dataset



(f) Convergence behavior of all algorithms using thyroid dataset



(g) Convergence behavior of all algorithms using vowel dataset



(h) Convergence behavior of all algorithms using LD dataset

Fig. 9 Convergence behaviour of compared algorithm using intra cluster distance parameter

Table 3 Average ranking of algorithms using Friedman test

Algorithm	K-Means	GA	PSO	ACO	CSO	Proposed algorithm
Ranking	5.13	4.81	3.25	3.44	3.13	1.25

Table 4 Statistical outcomes obtained through Friedman and Quade tests

Method	Statistical outcomes	Probability value (p)	Hypothesis	Critical value F (0.05,5, (35))	Critical value F (0.1,5, (35))
Friedman	22.096774	0.000502	R	11.070504	9.236361
Quade	9.649534	7.42E−6	R	2.485142	2.019125

Table 5 Relative size of each dataset for each algorithm using Quade test

Dataset/algorithm	K-Means	GA	PSO	ACO	CSO	Proposed algorithm
Iris	3	5	1	−1	−3	−5
Wine	6	10	−6	2	−2	−10
LD	7.5	−7.5	4.5	1.5	−1.5	−4.5
Cancer	12.5	5	−2.5	5	−7.5	−12.5
CMC	−2.5	2.5	0.5	−0.5	1.5	−1.5
Thyroid	17.5	10.5	3.5	−10.5	−3.5	−17.5
Glass	15	9	−9	−3	3	−15
Vowel	20	12	−12	4	−4	−20
Relative size	79	46.5	−20	−2.5	−17	−86

Table 6 Shows the ranking of algorithms using Quade test

Dataset/algorithm	K-Means	GA	PSO	ACO	CSO	Proposed algorithm
Iris	5	6	4	3	2	1
Wine	5	6	2	4	3	1
LD	6	1	5	4	3	2
Cancer	6	4.5	3	4.5	2	1
CMC	1	6	4	3	5	2
Thyroid	6	5	4	2	3	1
Glass	6	5	2	3	4	1
Vowel	6	5	2	4	3	1
Sum of Ranks	41	38.5	26	27.5	25	10
Avg. Ranking	5.13	4.81	3.25	3.44	3.13	1.25

Table 7 Results of Holm's Post-hoc test for Friedman test

i	Algorithms	z-values	*p* values	α/i, α = 0.05	Hypothesis	$\alpha/i, \alpha = 0.1$	Hypothesis
5	K-Means	5.8025	1.41E−06	0.0001	R	0.02	R
4	GA	5.3345	5.83E−06	0.0012	R	0.025	R
2	ACO	3.2756	0.002382	0.0166	R	0.033	R
3	PSO	2.9948	0.005016	0.025	R	0.05	R
1	CSO	2.8076	0.008104	0.05	R	0.1	R

Table 8 Results of Holm's Post-hoc test for Quade Test

i	Algorithms	z-values	*p* values	α/i, α = 0.05	Hypothesis	$\alpha/i, \alpha = 0.1$	Hypothesis
5	K-Means	5.9316	<.0001	0.0001	R	0.02	R
4	GA	5.2034	<.0001	0.0012	R	0.025	R
2	ACO	3.8249	0.003	0.0166	R	0.033	R
3	PSO	2.9243	0.0163	0.025	R	0.05	R
1	CSO	2.6492	0.0123	0.05	R	0.1	R

Table 9 Shows the ranking of algorithms using Friedman test

Algorithms	K-Means	GA	PSO	ACO	CSO	Proposed algorithm
Ranking	3.63	5.5	4	3.5	3.25	1.13

Table 10 Shows statistical results obtained through Friedman and Quade tests

Method	Statistical Outcomes	Probability value (p)	Hypothesis	Critical value F (0.05, 5, (35))	Critical value F (0.1,5, (35))
Friedman	23.115942	0.000321	R	11.070504	9.236361
Quade	9.583691	7.921E−6	R	2.485142	2.019125

Table 11 Shows the relative size of datasets using Quade test

Datasets/algorithm	K-Means	GA	PSO	ACO	CSO	Proposed algorithm
Iris	11.25	2.25	2.25	2.25	−6.75	−11.25
Wine	−6.75	11.25	6.75	2.25	−2.25	−11.25
LD	−11.25	11.25	2.25	6.75	−2.25	−6.75
Cancer	2.25	11.25	6.75	−6.75	−2.25	−11.25
CMC	−6.75	11.25	2.25	6.75	−2.25	−11.25
Thyroid	11.25	6.75	−2.25	−6.75	2.25	−11.25
Glass	−6.75	11.25	6.75	−2.25	2.25	−11.25
Vowel	11.25	6.75	−6.75	−2.25	2.25	−11.25
Relative Size of Datasets	4.5	72	18	0	−9	−85.5

Table 12 Shows the ranking of algorithms using Quade test

Dataset/algorithm	K-Means	GA	PSO	ACO	CSO	Proposed algorithm
Iris	6	4	4	4	2	1
Wine	2	6	5	4	3	1
LD	1	6	4	5	3	2
Cancer	4	6	5	2	3	1
CMC	2	6	4	5	3	1
Thyroid	6	5	3	2	4	1
Glass	2	6	5	3	4	1
Vowel	6	5	2	3	4	1
Sum of Ranks	29	44	32	28	26	9
Avg. ranking of algorithms	3.63	5.5	4	3.5	3.25	1.13

Table 13 Results of Holm's test for Friedman test

i	Algorithms	z-values	*p* values	α/i, α = 0.05	Hypothesis	α/i, α = 0.1	Hypothesis
5	GA	6.7826	7.32E−08	0.0001	R	0.02	R
4	PSO	4.4571	8.17E−05	0.0012	R	0.025	R
3	K-Means	3.8758	0.000447	0.0166	R	0.033	R
2	ACO	3.682	0.000775	0.025	R	0.05	R
1	CSO	3.0944	0.002264	0.05	R	0.1	R

Table 14 Results of Holm's test for Quade Test

i	Algorithms	z-values	*p* values	α/i, α = 0.05	Hypothesis	α/i, α = 0.1	Hypothesis
5	GA	5.9613	<.0001	0.0001	R	0.02	R
4	PSO	4.5184	0.0001	0.0012	R	0.025	R
3	K-Means	3.8947	0.0004	0.0166	R	0.033	R
2	ACO	3.4365	0.0008	0.025	R	0.05	R
1	CSO	2.8259	0.0023	0.05	R	0.1	R

References

1. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv 31(3):264–323
2. Lingras P, Huang X (2005) Statistical, evolutionary, and neuro-computing clustering techniques: cluster-based vs object-based approaches. Artif Intell Rev 23(1):3–29
3. Xu R, Wunsch D (2005) Survey of clustering algorithms. IEEE Trans Neural Netw 16(3):645–678
4. Chang DX, Zhang XD, Zheng CW (2009) A genetic algorithm with gene rearrangement for K-means clustering. Pattern Recognit 42:1210–1222
5. Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. KDD 96(34):226–231
6. Scheunders P (1997) A genetic c-means clustering algorithm applied to color image quantization. Pattern Recognit 30(6):859–866
7. Gomez-Muñoz VM, Porta-Gándara MA (2002) Local wind patterns for modeling renewable energy systems by means of cluster analysis techniques. Renew Energy 2:171–182
8. Mitra S, Banka H (2006) Multi-objective evolutionary bi clustering of gene expression data. Pattern Recognit 39:2464–2477
9. Cura T (2012) A particle swarm optimization approach to clustering. Expert Syst Appl 39(1):1582–1588
10. Kushwaha N, Pant M, Kant S, Jain VK (2018) Magnetic optimization algorithm for data clustering. Pattern Recognit Lett 115:59–65
11. Kumar Y, Sahoo G (2014) A charged system search approach for data clustering. Prog Artif Intell 2(2–3):153–166
12. Hatamlou A (2013) Black hole: a new heuristic optimization approach for data clustering. Inf Sci 222:175–184
13. Zhang C, Ouyang D, Ning J (2010) An artificial bee colony approach for clustering. Expert Syst Appl 37(7):4761–4767
14. Karaboga D, Ozturk C (2011) A novel clustering approach: artificial Bee Colony (ABC) algorithm. Appl Soft Comput 11(1):652–657
15. Runkler TA (2005) Ant colony optimization of clustering models. Int J Intell Syst 20(12):1233–1251
16. Erol OK, Eksin I (2006) A new optimization method: big bang–big crunch. Adv Eng Softw 37(2):106–111
17. Yang F, Sun T, Zhang C (2009) An efficient hybrid data clustering method based on K-harmonic means and Particle Swarm Optimization. Expert Syst Appl 36(6):9847–9852
18. Chuang LY, Hsiao CJ, Yang CH (2011) Chaotic particle swarm optimization for data clustering. Expert Syst Appl 38(12):14555–14563
19. Kumar Y, Sahoo G (2015) Hybridization of magnetic charge system search and particle swarm optimization for efficient data clustering using neighborhood search strategy. Soft Comput 19(12):3621–3645
20. Xue Y, Jiang J, Zhao B, Ma T (2018) A self-adaptive artificial bee colony algorithm based on global best for global optimization. Soft Comput 22(9):2935–2952
21. Huang CL, Huang WC, Chang HY, Yeh YC, Tsai CY (2013) Hybridization strategies for continuous ant colony optimization and particle swarm optimization applied to data clustering. Appl Soft Comput 13(9):3864–3872
22. Jordehi AR (2014) A chaotic-based big bang–big crunch algorithm for solving global optimisation problems. Neural Comput Appl 25(6):1329–1335
23. Nanda SJ, Panda G (2014) A survey on nature inspired metaheuristic algorithms for partitional clustering. Swarm Evol Comput 16:1–18
24. Tsai PW, Pan JS, Chen SM, Liao BY (2012) Enhanced parallel cat swarm optimization based on the Taguchi method. Expert Syst Appl 39(7):6309–6319
25. Ram G, Mandal D, Kar R, Ghoshal SP (2015) Cat swarm optimization as applied to time-modulated concentric circular antenna array: analysis and comparison with other stochastic optimization methods. IEEE Trans Antennas Propag 63(9):4180–4183
26. Pappula L, Ghosh D (2014) Linear antenna array synthesis using cat swarm optimization. AEU-Int J Electron Commun 68(6):540–549
27. Yusiong JPT (2012) Optimizing artificial neural networks using cat swarm optimization algorithm. Int J Intell Syst Appl 5(1):69
28. Bilgaiyan S, Sagnika S, Das M (2014) Workflow scheduling in cloud computing environment using cat swarm optimization. In: 2014 IEEE international advance computing conference (IACC). IEEE, pp 680–685
29. Temel S, Unaldi N, Kaynak O (2013) On deployment of wireless sensors on 3-D terrains to maximize sensing coverage by utilizing cat swarm optimization with wavelet transform. IEEE Trans Syst Man Cybern Syst 44(1):111–120

30. Guo L, Meng Z, Sun Y, Wang L (2016) Parameter identification and sensitivity analysis of solar cell models with cat swarm optimization algorithm. Energy Convers Manag 108:520–528
31. Kalaiselvan G, Lavanya A, Natrajan V (2011) Enhancing the performance of watermarking based on cat swarm optimization method. In: 2011 international conference on recent trends in information technology (ICRTIT). IEEE, pp 1081–1086
32. Cui SY, Wang ZH, Tsai PW, Chang CC, Yue S (2013) Single bitmap block truncation coding of color images using cat swarm optimization. In: Recent advances in information hiding and applications. Springer, Berlin, Heidelberg, pp 119–138
33. Chu SC, Tsai PW, Pan JS (2006) Cat swarm optimization. In: Pacific rim international conference on artificial intelligence. Springer Berlin Heidelberg, pp 854–858
34. Kumar Y, Singh PK (2018) Improved cat swarm optimization algorithm for solving global optimization problems and its application to clustering. Appl Intell 48(9):2681–2697
35. Pan JS, Dao TK, Kuo MY, Horng MF (2014) Hybrid bat algorithm with artificial bee colony. In: Intelligent data analysis and its applications, volume II. Springer, Cham, pp 45–55
36. Yan X, Zhu Y, Zou W, Wang L (2012) A new approach for data clustering using hybrid artificial bee colony algorithm. Neurocomputing 97:241–250
37. Siddiqi UF, Sait SM (2017) A new heuristic for the data clustering problem. IEEE Access 5:6801–6812
38. Kumar Y, Singh PK (2019) A chaotic teaching learning-based optimization algorithm for clustering problems. Appl Intell 49(3):1036–1062
39. Kumar V, Chhabra JK, Kumar D (2017) Grey wolf algorithm-based clustering technique. J Intell Syst 26(1):153–168
40. Boushaki SI, Kamel N, Bendjeghaba O (2018) A new quantum chaotic cuckoo search algorithm for data clustering. Expert Syst Appl 96:358–372
41. Kumar Y, Sahoo G (2017) Gaussian cat swarm optimisation algorithm based on Monte Carlo method for data clustering. Int J Comput Sci Eng 14(2):198–210
42. Jadhav AN, Gomathi N (2018) WGC: hybridization of exponential grey wolf optimizer with whale optimization for data clustering. Alex Eng J 57(3):1569–1584
43. Kumar A, Kumar D, Jarial S (2018) A novel hybrid K-means and artificial bee colony algorithm approach for data clustering. Decis Sci Lett 7(1):65–76
44. Lu J, Zhu Q, Wu Q (2018) A novel data clustering algorithm using heuristic rules based on k-nearest neighbors chain. Eng Appl Artif Intell 72:213–227
45. Huang F, Li X, Zhang S, Zhang J (2018) Harmonious genetic clustering. IEEE Trans Cybern 48(1):199–214
46. He H, Tan Y (2012) A two-stage genetic algorithm for automatic clustering. Neurocomputing 81:49–59
47. Orouskhani M, Mansouri M, Teshnehlab M (2011) Average-inertia weighted cat swarm optimization. In: International conference in swarm intelligence. Springer, Berlin, Heidelberg, pp 321–328
48. Mohapatra P, Chakravarty S, Dash PK (2016) Microarray medical data classification using kernel ridge regression and modified cat swarm optimization based gene selection system. Swarm Evol Comput 28:144–160
49. Lin KC, Huang YH, Hung JC, Lin YT (2015) Feature selection and parameter optimization of support vector machines based on modified cat swarm optimization. Int J Distrib Sens Netw 11(7):365869
50. Maulik U, Bandyopadhyay S (2000) Genetic algorithm-based clustering technique. Pattern Recognit 33(9):1455–1465
51. Van der Merwe DW, Engelbrecht AP (2003) Data clustering using particle swarm optimization. In: The 2003 congress on evolutionary computation, 2003, CEC'03. IEEE, vol 1, pp 215–220
52. Kumar Y, Sahoo G (2017) An improved cat swarm optimization algorithm based on opposition-based learning and cauchy operator for clustering. JIPS 13(4):1000–1013
53. Kumar Y, Sahoo G (2017) A two-step artificial bee colony algorithm for clustering. Neural Comput Appl 28(3):537–551
54. Wang ZH, Chang CC, Li MC (2012) Optimizing least-significant-bit substitution using cat swarm optimization strategy. Inf Sci 192:98–108