

A charged system search approach for data clustering

Yugal Kumar · G. Sahoo

Received: 11 December 2013 / Accepted: 16 March 2014 / Published online: 7 April 2014
© Springer-Verlag Berlin Heidelberg 2014

Abstract This paper presents a charged system search optimization method for finding the optimal cluster centers in a given dataset. CSS algorithm utilizes the Coulomb and Gauss laws from electrostatics to initiate the local search, and Newton second law of motion from mechanics is employed for global search. The efficiency and capability of the proposed algorithm are evaluated on seven datasets and compared with existing K -means, GA, PSO and ACO algorithms. From the experimental results, it is found that the proposed algorithm provides more accurate and effective results than other methods being compared.

Keywords Clustering · Coulomb law · Charge particles · Gauss law · Newton law

1 Introduction

The aim of clustering is to find out a subset of items in a given dataset which are more similar than others using similarity measures. The various authors have applied different criteria or similarity measures to identify the items in clusters. But the sum of the squared distances is widely accepted similarity measure for clustering problems. The cluster analysis has proven its significance in many areas such as pattern recognition [45,47], image processing [38], process monitoring [40], machine learning [1], quantitative structure activity relationship [9], document retrieval [18], bioinformatics [16], image

segmentation [34] and many more. Due to wide area of clustering in different domains, a large number of algorithms have been developed by various researchers and applied successfully for clustering. Generally, the clustering algorithms can be classified into two groups: hierarchical clustering algorithms and partition-based clustering algorithms [4,5,26,29]. In hierarchical algorithms, a tree structure of data is formed by merging or splitting data based on some similarity criterion. In partition-based algorithms, clustering is done by relocating data between clusters to the clustering criterion, i.e. Euclidian distance. From the literature, it has been found that partition-based algorithms are more efficient and popular than hierarchical algorithms [19]. The most popular and widely used partition-based algorithm is K -means algorithm. It is easy, fast, and simple to implement. In addition to it, there is also one more characteristic that is linear time complexity [11,20]. In K -means algorithm, a dataset is divided into K number of predefined clusters and used to minimize the intra-cluster distance based on Euclidean distance [19]. But, this algorithm has some limitations such as the results of K -means algorithm is highly dependent on the initial cluster centers and also get stuck in local minima [21,35]. Thus, to overcome the pitfalls of the K -means algorithm, several heuristic algorithms have been developed. K -harmonic mean algorithm has proposed for clustering instead of K -means in [46]. A simulated annealing (SA)-based approach has been developed in [36]. A tabu search (TS)-based method was introduced in [2,39]. A genetic algorithm (GA)-based methods were presented in [6,27,30,31]. Fathian et al. [10] have developed a clustering algorithm based on honey-bee mating optimization (HBMO). Shelokar et al. [37] proposed an ant colony optimization (ACO)-based approach for clustering. The particle swarm optimization (PSO) is applied for clustering in [44]. Hatamlou et al. employed a big bang-big crunch algorithm for data clustering in [13]. Karaboga and

Y. Kumar (✉) · G. Sahoo
Department of Information Technology, Birla Institute of Technology,
Mesra, Ranchi, Jharkhand, India
e-mail: yugalkumar.14@gmail.com

G. Sahoo
e-mail: gsahoo@bitmesra.ac.in

Ozturk presented a novel clustering approach based on artificial bee colony (ABC) algorithm in [23]. A data clustering based on gravitational search algorithm was presented in [14, 15]. But every algorithm has some drawbacks, for example, K -means algorithm sucks in local optima, convergence is highly dependent on initial positions in case of genetic algorithm; in ACO, the solution vector has been affected as the number of iterations increased, etc.

The aim of this research work is to explore the capability of charged system search (CSS) algorithm for data clustering. The CSS algorithm is the latest meta-heuristic optimization technique developed by Kaveh and Talatahari [24]. This technique is based on three principles: Coulomb law, Gauss law and Newton second law of motion. Every meta-heuristic algorithm contains two unique features, i.e. exploration and exploitation. The exploration is referred to generate the promising searching space, while the exploitation can be defined as determination of the most promising solution set. Thus in CSS, the exploration process is carried out using Coulomb and Gauss laws, while Newton second law of motion is applied to perform exploitation process. The performance of the proposed algorithm has been evaluated on two artificial datasets and several real datasets from UCI repository and compared with some existing algorithms in which quality of solution is improved using CSS algorithm.

2 CSS algorithm for clustering

In this section, CSS algorithm is explained to solve the clustering problem. The aim of this algorithm is to find out the optimal cluster points to assign N numbers of items to K cluster centers in R^n . In CSS algorithm, the sum of square of Euclidean distances is taken as the objective function for clustering problem and items are assigned to a cluster center with minimized Euclidean distance among cluster centers. The algorithm starts with defining the initial position and velocities of K number of charged particles (CPs). The initial positions of CPs are defined in a random manner. In CSS algorithm, CPs are assumed to be a charged sphere of radius ' a ' and its initial velocity is set to zero. Thus, the algorithm starts with randomly defined center points and ends with optimal cluster centers. Consider Table 1 which illustrates a dataset, used to explain the working of CSS algorithm for clustering with $N = 10$, $n = 4$ and the number cluster centers $K = 3$. To obtain the optimal cluster centers, the CPs use resultant electric force (attracting force vector), mass and moving probability of particles and cluster centers. After the first iteration, the velocities of CPs are determined and locations of CPs are also moved. The objective function is calculated again using the new positions of CPs and also compared with the old CPs position that are stored in memory pool, called as charged memory (CM). The CM is

Table 1 A dataset to explain the CSS algorithm for clustering with $N = 10$, $n = 4$ and $K = 3$

N	n	
	1	2
1	5.1	3.5
2	4.9	3
3	4.7	3.2
4	4.6	3.1
5	5	3.6
6	5.4	3.9
7	4.6	3.4
8	5	3.4
9	4.4	2.9
10	4.9	3.1

now updated with the new positions of CPs and excludes the worst CPs from CM. As the algorithm grows, the positions of CPs are updated along with the content in CM. This process progresses until the maximum iteration or no better position of CPs has been generated.

2.1 Algorithm details

As described earlier, the algorithm starts with the identification of the initial positions and velocities of CPs in random fashion. Thus, to identify the initial positions of randomly defined CPs (cluster centers), the given equation has been used to initialize the CPs position in original CSS. The modified equation can be given as.

$$C_k = X_{\min,i} + r_i \times (X_{i,\max} - X_{i,\min}),$$

where $i = 1, 2, \dots, n$ and $k = 1, 2, \dots, K$ (1)

In the above equation, C_k represents the number of cluster centers, r_i is a random function whose values lies between 0 and 1, $X_{\min,i}$ and $X_{\max,i}$ represent the minimum value and maximum value of the i th attribute of dataset; K is the total number of cluster centers in a dataset. The initial positions of CPs are given in Table 2.

It is assumed that the initial velocities of CPs are set to zero.

$$V_k = 0, \quad k = 1, 2, 3, \dots, K \quad (2)$$

In CSS algorithm, it is noted that the CPs are described as charged spheres. So, every CP contains some mass, and mass of each CP is calculated using the following equation.

$$m_k = \frac{\text{fit}(k) - \text{fit}(\text{worst})}{\text{fit}(\text{best}) - \text{fit}(\text{worst})} \quad (3)$$

where $\text{fit}(k)$ represents the fitness of k th instance of dataset, $\text{fit}(\text{best})$ represents the best fitness value and $\text{fit}(\text{worst})$ represents worst fitness value of dataset.

Table 2 Initial position of CPs using equation 1

K	i	
	1	2
1	4.6467	3.5664
2	4.4835	3.526
3	5.0609	3.6298

Table 3 Normalized value of objective function

N	K		
	1	2	3
1	0.45811	0.61706	0.1355
2	0.62046	0.67091	0.64999
3	0.37027	0.39132	0.56122
4	0.46875	0.44161	0.70222
5	0.35486	0.5218	0.067819
6	0.82382	0.9899	0.43358
7	0.17285	0.17159	0.51503
8	0.3905	0.53165	0.2377
9	0.71063	0.6315	0.98457
10	0.53074	0.59576	0.55366

The mass of initial positioned CPs is 0.91315, 0.70944 and 1.3907. The sum of the squared Euclidean distance is used as objective function in CSS algorithm to find the closeness of particles to CPs and assigned the particles to CPs with minimum objective value. Table 3 provides the value of objective function of initial positioned CPs for our example dataset. Euclidean distance can be given as

$$d_{i,k} = \sum_{k=1}^K \sum_{j=1}^N \sum_{i=1}^n \sqrt{\|X_{j,i} - C_{k,i}\|^2} \tag{4}$$

The information contained in given string is used to arrange the items into different clusters which are given below.

3	1	1	2	3	3	2	3	2	1
---	---	---	---	---	---	---	---	---	---

From the above string, it is observed that the first, fifth, sixth and eighth particles belong to the cluster third; second, third and tenth particles belong to cluster first and fourth, seventh and ninth particle belong to cluster second. Hence at this step, the dataset is divided into three different clusters and stores the values of positions of CPs in a new variable called charge memory (CM) which can be used to memorize the positions of CPs. Later on, these CPs position will be used for comparisons with newly generated CPs position and the best positions are included in the CM and excluding

the worst positions from CM. Here, the size of the CM is equal to the $N/4$. The main work of CM is to keep track of the number of good positions of CPs which are obtained during the execution of CSS algorithm and after the execution of algorithm; the optimal number of CPs position (i.e. K number of CPs) is determined using minimized objective function values. The above discussion relates to the initialization of CSS algorithm for clustering problem.

Now, we will describe the main steps of the CSS algorithm, i.e., how the new positions and velocities of CPs are to be generated. From the study of various meta-heuristic algorithms, it is found that every meta-heuristic algorithm contains two approaches, i.e. exploration and exploitation in which one approach initiated local search while the other approach carried out the global search. The local search tends to the exploration of random search space such that the most promising solution space can be occupied while the global search refers to the exploitation of good solution vectors from the promising solution space. Hence in case of CSS algorithm, the local search, i.e. exploration is initiated using Coulomb and Gauss laws while the global search, i.e. exploitation is performed by Newton second law of motion. The local search of CSS algorithm starts by measuring the electric force $E_{i,k}$ generated by CP. The electric force $E_{i,k}$ generated at a point either inside the CPs or outside CPs. So, this direction of electric force $E_{i,k}$ is described as moving probability ($P_{i,k}$) of CPs while the Coulomb and Gauss laws are applied to measure the total electric force generated on a CP, called actual electric force F_k . The moving probability $P_{i,k}$ for each CP can be determined using Eq. (5).

$$p_{ik} = \begin{cases} 1 & \text{if, } \frac{\text{fit}(i) - \text{fit}(\text{best})}{\text{fit}(k) - \text{fit}(i)} > \text{rand} \vee \text{fit}(k) > \text{fit}(i) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The value of moving probability $P_{i,k}$ is either 0 or 1 and it gives the information about the movement of CPs. Table 4 shows the moving probability $P_{i,k}$ values for each particle to each cluster center.

The Coulomb and Gauss laws are employed to determine the value of actual electric force F_k generated on CPs. The Coulomb law is used to calculate the force outside the CP and Gauss law is used to calculate the force inside the CP. The generalized equation to determine the actual electric force F_k can be given as:

$$F_k = q_k \sum_{i,i \neq k} \left(\frac{q_i}{a^3} \times i_1 + \frac{q_i}{r_{ik}^2} \times i_2 \right) \times p_{ik} \times (X_i - C_k),$$

$$\begin{cases} k = 1, 2, 3, \dots, K \\ i_1 = 1, i_2 = 0 \leftrightarrow r_{ik} < a \\ i_1 = 0, i_2 = 1 \leftrightarrow r_{ik} \geq a \end{cases} \tag{6}$$

Here, q_i and q_k represent the fitness of i th and k th CP, $r_{i,k}$ represents the separation distance between i th and k th CPs,

Table 4 Moving probability P_{ik} of each CP with each item of dataset

N	K		
	1	2	3
1	0	0	1
2	1	1	0
3	1	1	0
4	1	1	0
5	0	0	1
6	0	0	0
7	1	1	0
8	0	0	1
9	1	1	0
10	1	1	0

Table 5 Values of magnitude of charge (q_i) of each CP and separation distance ($r_{i,k}$)

N	q_i	Separation distance ($r_{i,k}$)		
		K		
		1	2	3
1	1.3	0.85265	1.5314	0.13776
2	0.6	0.79976	0.021312	1.7952
3	0.6	0.56345	0.42517	1.5859
4	0.4	1.2859	0.20939	2.2103
5	1.3	0.88357	1.6231	0.13828
6	2	1.6087	2.1519	0.72074
7	0.7	0.49457	0.5983	1.4444
8	1.1	0.54907	1.2877	0.47032
9	0.5	4.0248	4.0406	4.0154
10	0.7	0.39833	0.39626	1.4168

i_1 and i_2 are the two variables whose values are either 0 or 1, ‘ a ’ represents the radius of CPs and it is assumed that each CP has uniform volume charge density but changes in every iteration. The value of q_i , $r_{i,k}$ and ‘ a ’ is evaluated as follows:

$$q_i = \frac{\text{fit}(i) - \text{fit}(\text{worst})}{\text{fit}(\text{best}) - \text{fit}(\text{worst})} \quad i = 1, 2, 3, \dots, N \quad (7)$$

$$r_{ik} = \frac{\|X_i - X_k\|}{\|(X_i + X_k)/2 - X_{\text{best}}\| + \epsilon} \quad (8)$$

$$a = 0.10 \times \max(\{x_{i,\text{max}} - x_{i,\text{min}} | i = 1, 2, 3, \dots, n\}) \quad (9)$$

Tables 5, 6 and 7 provide the values of q_i , $r_{i,k}$, i_1 , i_2 and $(X_i - C_k)$ variables which are used to calculate the actual electric force F_k , applied on the CPs and the value of ‘ a ’ is 0.4647, 0.91302 and 1.4191 which is calculated using Eq. (9). The values of electric force F_k are used with Newton second law of motion to determine the new position of CPs and velocities of CPs.

Table 6 Values of I_1 and I_2

N	$K(I_1)$			$K(I_2)$		
	1	2	3	1	2	3
1	0	1	0	1	0	1
2	0	0	1	1	1	0
3	0	0	1	1	1	0
4	0	0	1	1	1	0
5	0	1	0	1	0	1
6	1	1	0	0	0	1
7	0	0	1	1	1	0
8	0	0	0	1	1	1
9	1	1	1	0	0	0
10	0	0	0	1	1	1

Table 7 Values of $X_i - C_k$ for each cluster center K

N	K		
	1	2	3
1	0.38685	0.59056	0.90696
2	0.31315	0.10944	0.7907
3	0.31315	0.10944	0.7907
4	0.51315	0.30944	0.9907
5	0.38685	0.59056	0.9269
6	1.0868	1.2906	0.6093
7	0.21315	0.9372	0.6907
8	0.18685	0.39056	0.2907
9	0.91315	0.70944	1.3907
10	0.21315	0.94426	0.6907

The actual electric force F_k generated on CPs is measured using above discussion and the values of F_k on initial CPs (cluster centers) are 0.85871, 0.29861 and 0.33496.

Newton second law of motion is employed to get the new positions and velocities of CPs. This is referred to as exploitation of solution vectors from the random space search. The new positions of CPs and velocities are obtained from Eqs. (10) and (11). Z_a and Z_v act as the control parameters which are used to control the exploration and exploitation process of CSS algorithm. These parameters also affect the values of previous velocities and actual resultant force generated on a CP. These values may be either increased or decreased. Thus, Z_a is the control parameter belonging to the actual electric force F_k and controls the exploitation process of CSS algorithm. The large value of Z_a increases the convergence speed of algorithm, while small value increases the computational time of algorithm. Z_v is the control parameter for exploration process and acts with the velocities of CPs. Here, it is noted that Z_a is the increased function parameter while Z_v is the decreased function parameter. Table 8 pro-

Table 8 New position of CPs

K	i	
	1	2
1	4.8604	3.78
2	4.5288	3.5305
3	5.174	3.6763

vides the new positions of CPs which are evaluated using CSS algorithm

$$C_{k,new} = \text{rand}_1 \times Z_a \times \frac{F_k}{m_k} * \Delta t^2 + \text{rand}_2 \times Z_v * V_{k,old} \times \Delta t + C_{k,old} \tag{10}$$

$$V_{k,new} = \frac{C_{k,new} - C_{k,old}}{\Delta t}, \tag{11}$$

where rand_1 and rand_2 are the two random functions whose values lie in between 0 and 1, Z_a and Z_v are the control parameters which control the influence of actual electric force and previous velocities, m_k is the mass of k th CPs which is equal to the q_k and Δt represents the time step which is set to 1.

The new positions of CPs are mentioned in Table 8 and the value of control parameters Z_a and Z_v is determined using Eq. (12). The new velocities ($V_{k,new}$) of each CP are 0.4273, 0.0498, 0.1596.

$$\begin{aligned} Z_a &= (1 - \text{iteration}/\text{iteration max}), \\ Z_v &= (1 + \text{iteration}/\text{iteration max}) \end{aligned} \tag{12}$$

Hence from the above discussion, the process of algorithm can be categorized into three sections: Initialization, Search and Termination condition. The initialization section deals with the CSS algorithm parameters; positions and velocities of initial CPs; determine the value of objective function and rank them; store the positions of CPs into CM. In the search section, the new positions and velocities of CPs are determined using moving probability $P_{i,k}$ and actual electric force F_k . The value of objective function is evaluated using newly generated CPs, compared with previous CPs; rank them and store the best CPs in CM. The termination condition of the algorithm is either maximum number of iterations or repeated positions of CPs. The flowchart of CSS algorithm for data clustering is depicted in Fig. 1.

3 Cluster analysis

The objective of the clustering algorithm is to group the similar data objects together. In literature, there are many techniques which have been employed in clustering analysis, such as partition-based clustering, hierarchical clustering,

density-based clustering, and artificial intelligence-based clustering. In this section, some artificial intelligence-based clustering methods have been described which have been used for comparing the results of our proposed algorithm. These are K -means, genetic algorithm (GA), PSO and ACO.

3.1 Application of K -means in clustering

K -means algorithm is one of the most popular and oldest methods developed by Macqueen [28] which has been widely applied in data clustering. It is one of the simple, fast and robust methods. In K -means algorithm, sum of Euclidean distance is used as similarity criteria to find the predefined number of cluster centers in a given dataset. K -means algorithm is started with randomly initialized cluster centers, and then the data vectors are arranged into predefined number of cluster cents according to minimum Euclidean distances. The cluster centers are updated using means of data vectors within cluster centers and this process is repeated until there is no improvement in cluster centers.

3.2 Applications of Genetic Algorithm (GA) in clustering

John Holland has proposed a genetic algorithm based on the Darwin theory of evolution [17] and has been applied in many function optimization problems. In GA, the string of chromosome describes the parameters of random search space. An objective function is associated with each string that defined the importance of string. Three operators are applied in GA to find the optimal solutions which are selection, crossover and mutation. To solve the clustering problem, Murthy and Chowdhury [31] have applied genetic algorithm-based method and evaluated the performance of using three experiments on synthetic and real-life data sets. The results conclude that the GA has improved the final outcome of K -means. Reported literature has shown that Al-Sultana and Maroof Khan [3] have studied several algorithms such as K -means algorithm, simulated annealing algorithm, tabu search algorithm, and genetic algorithm and compared the performance of these algorithms for the clustering problems. Krishna and Narasimha Murty [27] have proposed the GKA algorithm for clustering, and proved that the hybrid algorithm can converge to an optimal solution. Maulik and Bandyopadhyay [30] have applied the GA algorithm in clustering and evaluated the performance of this algorithm using four artificial and three real-life datasets. The results show that genetic algorithm is more superior to K -means algorithm. Tseng and Yang [42,43] have proposed a genetic algorithm-based approach which can automatically be used to find the proper number of clusters, and outcome of their algorithm superior than K -means.

Pseudo Code of CSS algorithm for clustering

Step 1: Load the dataset and specify the number of initial cluster centers (K).

Step 2: * initialize the initial positions and velocities of Charged Particles CPs * /
 For each charged particles $k=1$ to K /* K represents the total number of cluster centers */
 For each $j=1$ to m /* m represents the total number of features in dataset */
 Determine the value of initial position of charged particles (C_k) using equation 1;
 Calculate the value of mass for each C_k using equation 3;
 end
 end
 $V_k=0$;
 Iteration =0;

Step 3: Evaluate the value of objective function using sum of squared distances and assign the items to the clusters with minimum objective function value.

Step 4: Store the positions of initial charged particles (C_k) into a variable, called CM.

Step 5: while (the termination conditions are not met), do
 * Calculate the value of moving probability $P_{i,k}$ for each charged particle C_k * /
 For each charged particles $k=1$ to K
 For each $i=1$ to n
 Determine the value of fitness of each instance (q_{ik}) with each charged particles C_k using equation 5;
 end for
 if ($\text{fit}(q_{ik}) > \text{fit}(k)$)
 $P_{ik} \rightarrow 1$;
 else
 $P_{ik} \rightarrow 0$;
 end if
 end for

Step 6: * Determine the value of actual electric force F_k */
 Determine the value of mass for each instance q_i using equation 7.
 Calculate the value of radius 'a' using equation 9.
 For each charged particles $k=1$ to K
 Calculate the value of separation distance ($r_{i,k}$) using equation 8.
 If ($r_{i,k} < a$)
 $i_1 \rightarrow 1$
 else
 $i_2 \rightarrow 0$
 end if
 If ($r_{i,k} \geq a$)
 $i_2 \rightarrow 1$
 else
 $i_1 \rightarrow 0$
 end if
 end for
 For each charged particles $k=1$ to K
 Determine the value of actual electric force using equation 6.
 end for

Step 7: Calculate the new positions and velocities of charged particles using equation 10 and 11.

Step 8: Recalculate the value of objective function using new positions of charge particles.

Step 9: Compare the value of objective function of newly generated charge particles to charge particles reside in CM.

Step 10: Memorize the best solution achieved so far
 Iteration= Iteration +1;
 End while

Step 11: Output the best solution obtained.

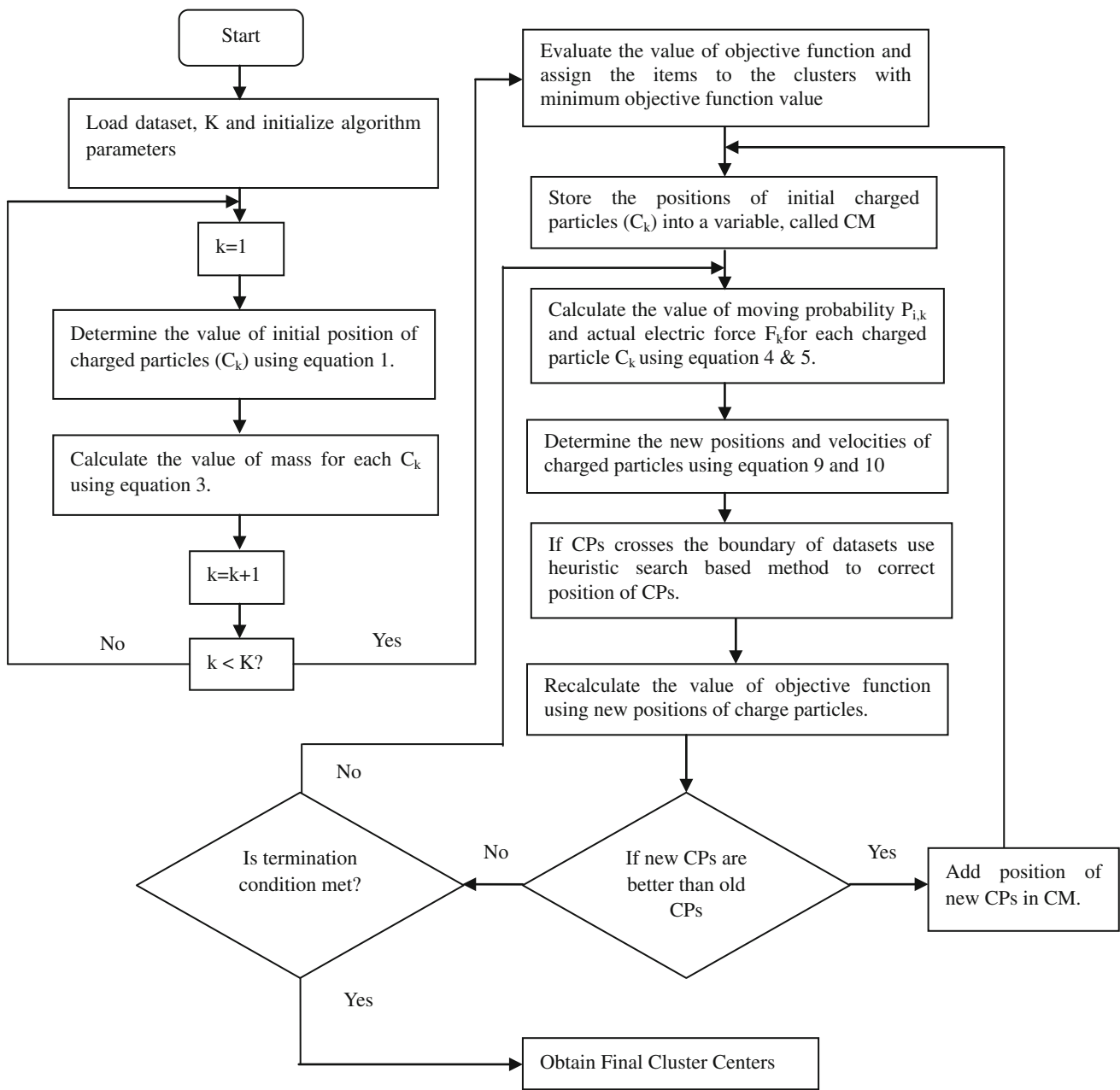


Fig. 1 Flowchart of CSS algorithm for data clustering

3.3 Applications of particle swarm optimization in clustering

Particle swarm optimization is a population-based stochastic search algorithm introduced in [25] and has been widely used to solve a broad range of optimization problems. This algorithm is based on social behavior of birds, insects, etc. and presented dispersion of individual swarm knowledge among all the members of group. Such as, if one member finds a desirable path to go, then rest of swarm will follow this path. In PSO, this behavior of animals is described by particles and each particle is associated with certain positions and veloci-

ties in a random search space. The algorithm is started using randomly initialized population. Each particle in PSO moves through the searching space and remembers the best position it has observed. Each particle shares the good positions to each other and updates their own position and velocity based on these good positions. The updation of velocity is based on the historical behaviors of the particles itself as well as their neighbors. Thus, the particles move towards better searching areas over the searching process. To investigate the performance of PSO algorithm, Van der Merwe and Engelbrecht [44] have applied the PSO algorithm for clustering in two ways—first, the PSO algorithm is used to obtain opti-

mal cluster centers for predefined number of cluster centers and second, PSO is used to refine the initial cluster centers for K -means algorithm. The results show that the PSO approaches have better convergence rate. A hybrid algorithm PSO-SA has been developed to obtain good cluster partition in [32]. In this algorithm, SA is applied to obtain global solution in PSO and the proposed algorithm has provided optimal solution. Niknam and Amiri [33] have proposed a hybrid approach based on PSO, SA and K -means for cluster analysis and the experimental results show that the proposed approach has obtained better result to PSO, SA, PSO-SA and K -means.

3.4 Applications of ant colony optimization in clustering

Ant colony optimization is a meta-heuristic algorithm proposed by Dorigo et al. [8] for combinatorial optimization problems. This algorithm simulates the behavior of real ants, i.e. how ants find the shortest path between food sources to nest. In ACO, artificial ants are used to construct the solutions by traversing the fully connected graph $G(V, E)$, where V is a set of vertices and E is a set of edges. Each artificial ant moves from vertex to vertex along the edges of the graph and constructs a partial solution. However, each ant also deposits a certain amount of pheromone on each vertex that they have traversed. The amount of pheromone deposit depends on the quality of the solution obtain. Ants have used the pheromone information to find out the promising regions in the search space. To inspect the efficiency of ACO algorithm in clustering domain, an ant-based clustering approach has been proposed in [37]. The simulation results indicate that the proposed algorithm has provided better results in terms of quality of solutions. Kao et al. [22] have proposed an ACO-based algorithm for clustering and named it ACOC. The performance of ACOC algorithm is compared with K -means and Shelokar ACO algorithm in which ACOC has given better results. Tsai et al. [41] have applied the ant system with different favorable strategy for data clustering and named as ACO with different favor (ACODF). ACODF algorithm has the following desirable strategies. It first uses differently favorable ants to solve the clustering problem. Then, the proposed ant colony system adopted the simulated annealing concept for ants to decrease the visit to the number of cities and get the local optimal solutions.

4 Experimental results

This section describes the results of the CSS algorithm for data clustering problem. To assess the performance of CSS algorithm, it is applied on ten datasets. These datasets are ART1, ART2, iris, wine, CMC, glass, breast cancer Wisconsin, liver disease (LD), thyroid and vowel in which iris, wine,

Table 9 Characteristics of datasets

Dataset	Classes	Features	Total instances	Instance in each class
ART 1	3	2	300	(100, 100, 100)
ART 2	3	3	300	(100, 100, 100)
Iris	3	4	150	(50, 50, 50)
Glass	6	9	214	(70, 17, 76, 13, 9, 29)
LD	2	6	345	(145, 200)
Thyroid	3	3	215	(150, 30, 35)
Cancer	2	9	683	(444, 239)
CMC	3	9	1473	(629, 334, 510)
Vowel	6	3	871	(72, 89, 172, 151, 207, 180)
Wine	3	13	178	(59, 71, 48)

Table 10 Parameters setting for CSS algorithm

Parameters	Value
No. of CPs	No. of clusters
rand	Random value between [0,1]
c	0.1
ϵ	0.001

CMC, glass, liver disease (LD), thyroid, vowel and breast cancer Wisconsin datasets are real that are downloaded from UCI repository while rest of the two datasets are artificial, i.e., ART1 and ART2. The characteristics of these datasets are discussed in Table 9. The proposed algorithm is implemented in Matlab 2010a environment on a core i5 processor with 4 GB using window operating system. For every dataset, the algorithm runs 20 times individually to check the effectiveness of proposed algorithm using randomly generated cluster centers. The parameter settings for CSS algorithm are mentioned in Table 10. The sum of intra-cluster distances and f -measure is used to evaluate the quality of solutions for clustering algorithm. The sum of intra-cluster distances can be defined as distances between the instances placed in a cluster to the corresponding cluster center. The results are measured in terms of best case, average cases, worst case solutions and standard deviation. The quality of clustering is directly related to the minimum sum of distances. The accuracy of clustering is measured using f -measure. To ensure the effectiveness and adaptability of CSS algorithm in clustering domain, the investigational results of CSS algorithm are compared with K -means, GA, PSO and ACO algorithms which are given in Table 11.

4.1 Datasets

4.1.1 ART1

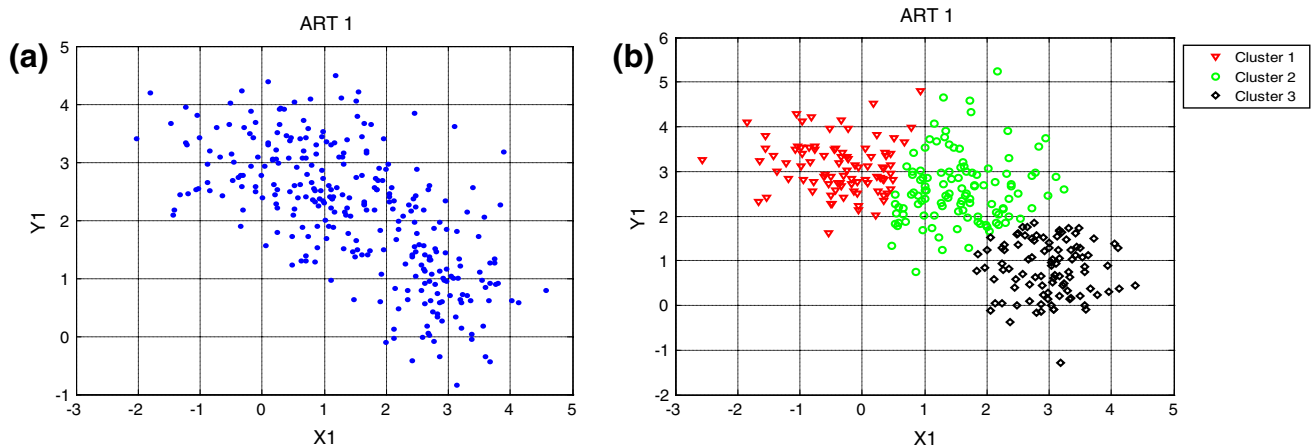
It is a two-dimensional artificial dataset, generated in matlab to authenticate the proposed algorithm. This dataset includes

Table 11 Comparisons of different clustering algorithm with CSS algorithm

Dataset	Parameters	<i>K</i> -means	GA	PSO	ACO	CSS
ART 1	Best	157.12	154.46	154.06	154.37	153.91
	Average	161.12	158.87	158.24	158.52	158.29
	Worst	166.08	164.08	161.83	162.52	161.32
	Std	0.34	0.281	0	0	0
	<i>F</i> -measure	99.14	99.78	100	100	100
ART2	Best	743	741.71	740.29	739.81	738.96
	Average	749.83	747.67	745.78	746.01	745.61
	Worst	754.28	753.93	749.52	749.97	749.66
	Std	0.516	0.356	0.237	0.206	0.209
	<i>F</i> -measure	98.94	99.17	99.26	99.19	99.43
Iris	Best	97.33	113.98	96.89	97.1	96.47
	Average	106.05	125.19	97.23	97.17	96.63
	Worst	120.45	139.77	97.89	97.8	96.78
	Std	14.631	14.563	0.347	0.367	0.14
	<i>F</i> -measure	0.782	0.778	0.782	0.779	0.787
Wine	Best	16,555.68	16,530.53	16,345.96	16,530.53	16,282.12
	Average	18,061	16,530.53	16,417.47	16,530.53	16,289.42
	Worst	18,563.12	16,530.53	16,562.31	16,530.53	16,317.67
	Std	793.213	0	85.497	0	10.31
	<i>F</i> -measure	0.521	0.515	0.518	0.519	0.529
LD	Best	11,397.83	532.48	209.15	224.76	207.09
	Average	11,673.12	543.69	224.47	235.16	228.27
	Worst	12,043.12	563.26	239.11	256.44	242.14
	Std	667.56	41.78	29.38	17.46	18.54
	<i>F</i> -measure	0.467	0.482	0.493	0.487	0.491
Cancer	Best	2,999.19	2,999.32	2,973.5	2,970.49	2,946.48
	Average	3,251.21	3,249.46	3,050.04	3,046.06	2,961.16
	Worst	3,521.59	3,427.43	3,318.88	3,242.01	3,006.14
	Std	251.14	229.734	110.801	90.5	12.23
	<i>F</i> -measure	0.829	0.819	0.819	0.821	0.847
CMC	Best	5,842.2	5,705.63	5,700.98	5,701.92	5,672.46
	Average	5,893.6	5,756.59	5,820.96	5,819.13	5,687.82
	Worst	5,934.43	5,812.64	5,923.24	5,912.43	5,723.63
	Std	47.16	50.369	46.959	45.634	21.43
	<i>F</i> -measure	0.334	0.324	0.331	0.328	0.359
Thyroid	Best	13,956.83	10,176.29	10,108.56	10,085.82	9,997.25
	Average	14,133.14	10,218.82	10,149.7	10,108.13	10,078.23
	Worst	146,424.21	10,254.39	10,172.86	10,134.82	10,116.52
	Std	246.06	32.64	27.13	21.34	49.02
	<i>F</i> -measure	0.731	0.763	0.778	0.783	0.789
Glass	Best	215.74	278.37	270.57	269.72	203.58
	Average	235.5	282.32	275.71	273.46	223.44
	Worst	255.38	286.77	283.52	280.08	241.27
	Std	12.47	4.138	4.55	3.584	13.29
	<i>F</i> -measure	0.431	0.333	0.359	0.364	0.446

Table 11 continued

Dataset	Parameters	<i>K</i> -means	GA	PSO	ACO	CSS
Vowel	Best	149,422.26	149,513.73	148,976.01	149,395.6	149,335.61
	Average	159,242.89	159,153.49	151,999.82	159,458.14	152,128.19
	Worst	161,236.81	165,991.65	158,121.18	165,939.82	154,537.08
	Std	916	3,105.544	2,881.346	3,485.381	2,128.023
	<i>F</i> -measure	0.652	0.647	0.648	0.649	0.649

**Fig. 2** a Distribution of data in ART1. b Clustered the ART1 data using CSS

300 instances with the two attributes and three classes. The classes in dataset are disseminated using μ and λ , where μ is the mean vector and λ is the variance matrix. The data have generated using $\mu_1 = [3, 1]$, $\mu_2 = [0, 3]$, $\mu_3 = [1.5, 2.5]$ and $\lambda_1 = [0.3, 0.5]$, $\lambda_2 = [0.7, 0.4]$, $\lambda_3 = [0.4, 0.6]$. Figure 2a depicts the distribution of data into ART1 and Fig. 2b shows the clustering of same data using CSS method.

4.1.2 ART2

It is three-dimensional artificial data which include 300 instances with three attributes and three classes. The data have generated using $\mu_1 = [10, 25, 12]$, $\mu_2 = [11, 20, 15]$, $\mu_3 = [14, 15, 18]$ and $\lambda_1 = [3.4, -0.5, -1.5]$, $\lambda_2 = [-0.5, 3.2, 0.8]$, $\lambda_3 = [-1.5, 0.1, 1.8]$. Figure 3a represents the distribution of data in ART2 dataset and Fig. 3b, c shows the clustering of same data using CSS method.

4.1.3 Iris dataset

Iris dataset contains three varieties of the iris flowers which are setosa, versicolour and virginica. The dataset contains 150 instances with three classes and four attributes in which each class contains 50 instances. The attributes of iris

dataset are sepal length, sepal width, petal length, and petal width.

4.1.4 Wine dataset

It contains the chemical analysis of wine in the same region of Italy using three different cultivators. This dataset contains 178 instances with thirteen attributes and three classes. The attributes of dataset are alcohol, malic acid, ash, alkalinity of ash, magnesium, total phenols, flavanoids, nonflavanoid phenols, proanthocyanins, color intensity, hue, OD280/OD315 of diluted wines and proline.

4.1.5 Glass

This dataset consists of six different types of glass information. The dataset contains 214 instances and 7 classes. It contains nine attributes which are refractive index, sodium, magnesium, aluminium, silicon, potassium, calcium, barium, and iron.

4.1.6 Breast cancer Wisconsin

This dataset characterizes the behavior of cell nuclei present in the image of breast mass. It contains 683 instances with

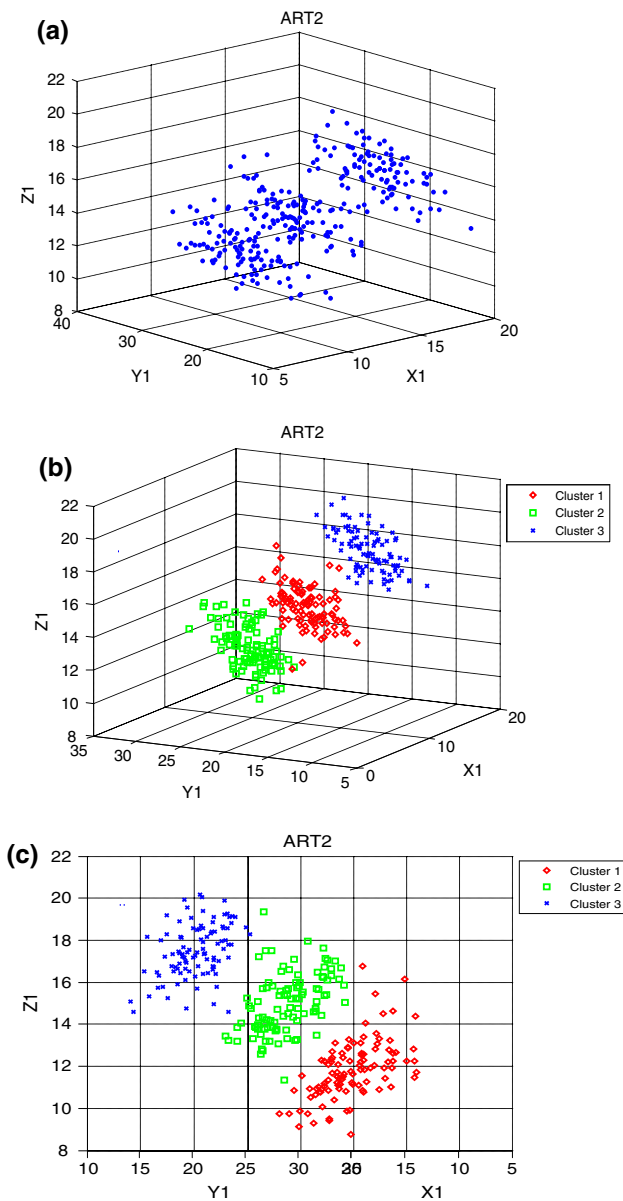


Fig. 3 a Distribution of data in ART2. b Clustered the ART2 data using CSS (*horizontal view*). c Clustered the ART2 data using CSS (*vertical view* as X, Y co-ordinate are in *horizontal plane* and Z coordinate in *vertical plane*)

2 classes, i.e. malignant and benign and 9 attributes. The attributes are clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. Malignant class consists of 444 instances while benign consists of 239 instances.

Table 12 Cluster center generated using CSS method for ART1 and ART2 dataset

Dataset	Center 1	Center 2	Dataset	Center 1	Center 2	Center 3
ART1	0.702	3.1316	ART2	10.7854	19.9337	15.4014
	1.8309	2.4381		14.3319	14.8893	17.5202
	2.9439	0.9189		9.7102	25.1293	12.404

4.1.7 Contraceptive method choice

It is a subset of National Indonesia Contraceptive Prevalence Survey data that had been performed in 1987. This dataset contains the information about married women who were either pregnant (but did not know about pregnancy) or not pregnant. It contains 1,473 instances and three classes, i.e., no use, long-term method and short term method. Each class contains 629, 334 and 510 instances, respectively. It has nine attributes which are Age, Wife’s education, Husband’s education, Number of children ever born, Wife’s religion, Wife’s now working, Husband’s occupation, Standard-of-living index and Media exposure.

4.1.8 Thyroid

This dataset contains the information about the thyroid diseases and classifies the patient into three classes—normal, hypothyroidism and hyperthyroidism. The dataset consist of 215 instances with five features. The features are the medical tests which have been used to categorize the patients. The features are T3resin, Thyroxin, Triiodothyronine, Thyroid stimulating and TSH value.

4.1.9 Liver disorder

This dataset is collected by BUPA medical research company. It consists of 345 instances with six features and two classes. The features of the LD dataset are mcv, alkphos, sgpt, sgot, gammagt and drinks.

4.1.10 Vowel

This dataset consists of 871 data instances of Indian Telugu vowel sounds with three features which correspond to the first, second, and third vowel frequencies and six classes.

4.2 Performance measures

4.2.1 Sum of intra-cluster distances

It is sum of distances between the data instances present in one cluster to its corresponding cluster center. Minimum sum of intra-cluster distance indicates the better the quality of the solution. The results are measured in terms of best, average and worst solutions.

Table 13 Cluster center of Iris, Wine and CMC dataset using CSS algorithm

Dataset	Center 1	Center 2	Center 3	Dataset	Center 1	Center 2	Center 3
Iris	5.01192	5.92342	6.73197	Wine	13.72460	12.50948	12.80496
	3.39790	2.78650	3.05658		1.85983	2.32095	2.53956
	1.46236	4.40893	5.63654		2.43126	2.32928	2.38043
	0.23236	1.40972	2.11840		16.92021	21.32354	19.50441
CMC	24.41967	33.50716	43.64541	105.27590	92.53096	98.93987	
	3.03923	3.13165	2.99234	2.85793	2.03432	2.06126	
	3.51074	3.55079	3.44623	3.06254	1.77883	1.49214	
	1.78976	3.65913	4.59062	0.29317	0.40460	0.42674	
	0.92598	0.78914	0.79621	2.01613	1.43797	1.41763	
	0.79163	0.69682	0.76526	5.69821	4.35484	5.77860	
	2.29435	2.10172	1.82611	1.07763	0.95039	0.88806	
	2.97130	3.28469	3.42527	3.02412	2.46158	2.22246	
	0.03734	0.05985	0.09168	1137.27300	463.59910	686.96590	

Table 14 Cluster center of Glass dataset using CSS algorithm

Dataset	Center 1	Center 2	Center 3	Center 4	Center 5	Center 6
Glass	1.52092	1.52177	1.51703	1.5147	1.51483	1.53005
	13.72637	13.10112	13.08151	14.63508	13.82799	16.27164
	3.47891	0.25012	3.52873	0.06379	2.41686	3.74163
	1.03876	1.42751	1.36193	2.21252	2.62295	2.9632
	71.91316	72.6832	72.85928	73.26839	71.11189	74.47665
	0.22203	0.30306	0.57581	0.04828	2.57717	5.1734
	9.45223	11.97474	8.37125	8.6934	5.97278	14.39661
	0.03526	0.05002	0.003921	1.00563	1.3349	2.6236
	0.05577	0.05656	0.05811	0.01843	0.00001	0.42456

Table 15 Cluster center of Cancer dataset using CSS algorithm

Cluster center	Cancer								
Center 1	2.88692	1.12763	1.20059	1.16407	1.99241	1.12131	2.00595	1.10154	1.03163
Center 2	7.11638	6.64189	6.62571	5.61392	5.24078	8.10028	6.07673	6.02127	2.32726

4.2.2 Standard deviation

Standard deviation provides the information about the dispersion of data instances present in cluster from its cluster center. The minimum value of standard deviation indicates that the data instances are close to its center, while large value indicates that the data are far from its center points.

4.2.3 F-measure

F-measure is calculated by the recall and precision of an information retrieval system [7, 12]. It is weighted harmonic mean of recall and precision. To determine the value of f-measure, every cluster describes a result of query while

Table 16 Cluster center of Thyroid dataset using CSS algorithm

Cluster center	Thyroid				
Center 1	0.9257	-1.1237	-0.4364	0.9831	4.1493
Center 2	1.693	-1.6712	-1.0296	2.3142	0.8312
Center 3	-0.1786	0.1584	0.2624	-0.2978	-0.2169

every class describes a set of credentials for query. Thus, if each cluster j consists of a set of n_j data instances as a result of a query and each class i consists of a set of n_i data instances required for a query then n_{ij} gives the number of instances of class i within cluster j . The recall and precision, for each cluster j and class i are defined as:

Table 17 Cluster center of Vowel dataset using CSS algorithm

Dataset	Center 1	Center 2	Center 3	Center 4	Center 5	Center 6
Vowel	506.75791	407.4528	624.8965	356.37525	376.42763	437.28721
	1,839.45312	1,012.05152	1,308.7234	2,290.97653	2,152.01786	991.65418
	2,556.18164	2,311.126	2,333.29831	2,977.08562	2,677.36256	2,658.21397

Table 18 Cluster center of LD dataset using CSS algorithm

Dataset	Center 1	Center 2	Center 3	Center 4	Center 5	Center 6
LD	87.86525	69.64812	26.10357	22.24653	26.91362	2.91364
	91.02736	74.97367	59.82641	39.17032	130.00263	5.93261

$$\text{Recall } (r(i, j)) = \frac{n_{i,j}}{n_i} \text{ and Precision } (p(i, j)) = \frac{n_{i,j}}{n_j} \tag{13}$$

The value of *F*-measure (*F*(*i, j*)) is computed as

$$F(i, j) = \frac{2 \times (\text{Recall} \times \text{Precision})}{(\text{Recall} + \text{Precision})} \tag{14}$$

Finally, the value of *F*-measure for a given clustering algorithm which consists of *n* number of data instances is given as

$$F(i, j) = \sum_{i=1}^n \frac{n_i}{n} \max_i (F(i, j)) \tag{15}$$

From Table 11, it can be seen that the results obtained from the CSS algorithm are better as compared with the other algorithms. The best values achieved by the algorithm for iris, wine, cancer, CMC, glass, LD, thyroid and vowel datasets are 96.47, 16,282.12, 2,946.48, 5,679.46, 223.58, 207.09, 9,997.25 and 149,535.61. The CSS algorithm gives better results with iris, wine, cancer, CMC, glass, and thyroid datasets while for the LD and vowel datasets, PSO algorithm gives better performance than CSS algorithm. But from the simulation results, it is observed that CSS algorithm obtains minimum value of best distance parameter for the LD dataset and worst distance parameter for vowel dataset among all methods being compared. The standard deviation parameter shows how much the data are far from the cluster centers. The value of standard deviation parameter for CSS algorithm is also smaller than the other methods. Moreover, the CSS algorithm provides better *f*-measure values than others which show higher accuracy of the said algorithm. To prove the viability of the results given in Table 11, the best centers obtained by the CSS algorithm are given in Tables 12, 13, 14, 15, 16, 17 and 18.

5 Conclusion

In this paper, a CSS algorithm is applied to solve the clustering problem. In the proposed algorithm, Newton second law

of motion is used to get the optimal cluster centers but it is the actual electric force *F_k* which plays a vital role to obtain the optimal cluster centers. Hence, the working of proposed algorithm is divided into two steps, first step involves calculation of the value of actual electric force using Coulomb and Gauss laws. In second step, the optimal cluster centers using Newton second law of motion are obtained. The CSS algorithm can be applied for data clustering when number of cluster centers (*K*) is already known. The performance of the CSS algorithm is tested on the several datasets and compared with *K*-means, GA, PSO and ACO, in which proposed algorithm provides better results and the quality of solutions obtained by the proposed algorithm is found to be superior in comparison to the other algorithms.

References

- Alpaydin, E.: Introduction to Machine Learning. MIT press, Cambridge (2004)
- Al-Sultana, K.S.: A Tabu search approach to the clustering problem. Pattern Recognit. **28**, 1443–1451 (1995)
- Al-Sultana, K.S., Maroof Khan, M.: Computational experience on four algorithms for the hard clustering problem. Pattern Recognit. Lett. **17**(3), 295–308 (1996)
- Barbakh, W., Wu, Y., Fyfe, C.: Review of Clustering Algorithms. Non-Standard Parameter Adaptation for Exploratory Data Analysis. Springer, Berlin (2009)
- Camastra, F., Vinciarelli, A.: Clustering Methods. Machine Learning for Audio, Image and Video Analysis. Springer, London, pp. 117–148 (2008)
- Cowgill, M.C., Harvey, R.J., Watson, L.T.: A genetic algorithm approach to cluster analysis. Comput. Math. Appl. **37**, 99–108 (1999)
- Dalli, A.: Adaptation of the F-measure to cluster based lexicon quality evaluation. In: Proceedings of the EACL 2003, pp. 51–56. Association for Computational Linguistics (2003)
- Dorigo, M., Maniezzo, V., Colomi, A.: Ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. Part B Cybern. **26**(1), 29–41 (1996)
- Dunn III, W.J., Greenberg, M.J., Callejas, S.S.: Use of cluster analysis in the development of structure–activity relations for antitumor triazines. J. Med. Chem. **19**(11), 1299–1301 (1976)

10. Fathian, M., Amiri, B., Maroosi, A.: Application of honey-bee mating optimization algorithm on clustering. *Appl. Math. Comput.* **190**, 1502–1513 (2007)
11. Forgy, E.W.: Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics* **21**, 2 (1965)
12. Handl, J., Knowles, J., Dorigo, M.: On the performance of ant-based clustering. *Design Appl. Hybrid Intell. Syst. Front. Artif. Intell. Appl.* **104**, 204–213 (2003)
13. Hatamlou, A., Abdullah, S., Hatamlou, M.: Data clustering using big bang-big crunch algorithm, pp. 383–388. *Communications in Computer and Information, Science*(2011)
14. Hatamlou, A., Abdullah, S., Nezamabadi-pour, H.: Application of Gravitational Search Algorithm on Data Clustering. *Rough Sets and Knowledge Technology*. Springer, Berlin (2011)
15. Hatamlou, A., Abdullah, S., Nezamabadi-pour, H.: A combined approach for clustering based on K-means and gravitational search algorithms. *Swarm Evol. Comput.* **6**, 47–52 (2012)
16. He, Y., Pan, W., Lin, J.: Cluster analysis using multivariate normal mixture models to detect differential gene expression with microarray data. *Comput. Stat. Data Anal.* **51**(2), 641–658 (2006)
17. Holland, J.H.: Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. Michigan Press, Michigan (1975)
18. Hu, G., Zhou, S., Guan, J., Hu, X.: Towards effective document clustering: a constrained k -means based approach. *Inf. Process. Manage.* **44**(4), 1397–1409 (2008)
19. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. *ACM Computing Surveys*, pp. 264–323 (1999)
20. Jain, A.K.: Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* **31**, 651–666 (2010)
21. Kao, Y.-T., Zahara, E., Kao, I.W.: A hybridized approach to data clustering. *Expert Syst. Appl.* **34**, 1754–1762 (2008)
22. Kao, Y., Cheng, K.: An ACO-based clustering algorithm. In: *Ant Colony Optimization and Swarm Intelligence*, pp. 340–347. Springer, Berlin (2006)
23. Karaboga, D., Ozturk, C.: A novel clustering approach: Artificial Bee Colony (ABC) algorithm. *Appl. Soft Comput.* **11**(1), 652–657 (2011)
24. Kaveh, A., Talatahari, S.: A novel heuristic optimization method: charged system search. *Acta Mech.* **213**(3–4), 267–289 (2010)
25. Kennedy, J., Eberhart, R.C.: A discrete binary version of the particle swarm algorithm. *IEEE Int. Conf. Syst. Man Cybern. Comput. Cybern. Simul.* **5**, 4104–4108 (1997)
26. Kogan, J., Nicholas, C., Teboulle, M., Berkhin, P.: A Survey of Clustering Data Mining Techniques. *Grouping Multidimensional Data*. Springer, Berlin (2006)
27. Krishna, K., Narasimha Murty, M.: Genetic K-means algorithm. *IEEE Trans. Syst. Man Cybernet. Part B Cybern.* **29**, 433–439 (1999)
28. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 281–297, p. 14 (1967)
29. Maimon, O., Rokach, L.: A survey of Clustering Algorithms. *Data Mining and Knowledge Discovery Handbook*. Springer, US (2010)
30. Maulik, U., Bandyopadhyay, S.: Genetic algorithm-based clustering technique. *Pattern Recognit.* **33**, 1455–1465 (2000)
31. Murthy, C.A., Chowdhury, N.: In search of optimal clusters using genetic algorithms. *Pattern Recognit. Lett.* **17**, 825–832 (1996)
32. Niknam, T., Amiri, B., Olamaei, J., Arefi, A.: An efficient hybrid evolutionary optimization algorithm based on PSO and SA for clustering. *J. Zhejiang Univ. Sci. A* **10**(4), 512–519 (2009)
33. Niknam, T., Amiri, B.: An efficient hybrid approach based on PSO, ACO and k -means for cluster analysis. *Appl. Soft Comput.* **10**(1), 183–197 (2010)
34. Pappas, T.N.: An adaptive clustering algorithm for image segmentation. *IEEE Trans. Signal Process.* **40**(4), 901–914 (1992)
35. Selim, S.Z., Ismail, M.A.: K-means-type algorithms: a generalized convergence theorem and characterization of local optimality. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 81–87 (1984)
36. Selim, S.Z., Alsultan, K.: A simulated annealing algorithm for the clustering problem. *Pattern Recognit.* **24**, 1003–1008 (1991)
37. Shelokar, P.S., Valadi, K., Jayaraman, Kulkarni, B.D.: An ant colony approach for clustering. *Anal. Chim. Acta* **509**(2), 187–195 (2004)
38. Sonka, M., Hlavac, V., Boyle, R.: *Image processing, analysis, and machine vision* (1999)
39. Sung, C.S., Jin, H.W.: A tabu-search-based heuristic for clustering. *Pattern Recognit.* **33**, 849–858 (2000)
40. Teppola, P., Mujunen, S.-P., Minkkinen, P.: Adaptive fuzzy C-means clustering in process monitoring. *Chemometr. Intell. Lab. Syst.* **45**(1), 23–38 (1999)
41. Tsai, C.-F., Tsai, C.-W., Yang, T.: ACODF: a novel data clustering approach for data mining in large databases. *J. Syst. Softw.* **73**(1), 133–145 (2004)
42. Lin, Y., Tseng, S.B.: Genetic algorithms for clustering feature selection and classification. *IEEE Int. Conf. Neural Netw.* **3**, 1612–1616 (1997)
43. Tseng, L.Y., Yang, S.B.: A genetic approach to the automatic clustering problem. *Pattern Recognit.* **34**(2), 415–424 (2001)
44. Van der Merwe, D.W., Engelbrecht, A.P.: Data clustering using particle swarm optimization. In: *Congress on Evolutionary Computation CEC-03*, vol. 1, pp. 215–220. IEEE (2003)
45. Webb, A.: *Statistical Pattern Recognition*. New Jersey: Wiley, pp. 361–406 (2002)
46. Zhang, B., Hsu, M., Dayal, U.: K-Harmonic means—a data clustering algorithm. *Hewlett-Packard Labs Technical Report HPL* (1999)
47. Zhou, H., Liu, Y.: Accurate integration of multi-view range images using k -means clustering. *Pattern Recognit.* **41**(1), 152–175 (2008)