

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



A Clustering Approach Based on Charged Particles

Yugal Kumar, Sumit Gupta,
Dharmender Kumar and Gadadhar Sahoo

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/63081>

Abstract

In pattern recognition, clustering is a powerful technique that can be used to find the identical group of objects from a given dataset. It has proven its importance in various domains such as bioinformatics, machine learning, pattern recognition, document clustering, and so on. But, in clustering, it is difficult to determine the optimal cluster centers in a given set of data. So, in this paper, a new method called magnetic charged system search (MCSS) is applied to determine the optimal cluster centers. This method is based on the behavior of charged particles. The proposed method employs the electric force and magnetic force to initiate the local search while Newton second law of motion is employed for global search. The performance of the proposed algorithm is tested on several datasets which are taken from UCI repository and compared with the other existing methods like K-Means, GA, PSO, ACO, and CSS. The experimental results prove the applicability of the proposed method in clustering domain.

Keywords: clustering, charged particles, electric force, magnetic force, Newton Law

1. Introduction

Clustering is an unsupervised technique which can be applied to understand the organization of data. The basic principle of clustering is to partition a set of objects into a set of clusters such that the objects within a cluster share more similar characteristics in comparison to the other clusters. A pre-specified criterion has been used to measure the similarity between the objects. In clustering, there is no need to train the data, it only deals with the internal structure of data and used a similarity criterion to group the objects into different clusters. Due to this, it is also known as unsupervised classification technique. It becomes a NP hard problem when number of clusters is greater than three. Consider a set $S = [A_1, A_2, A_3 \dots A_N]$ such that $A_i \in S$, consist

of N number of data objects and another set $P = [B_1, B_2 \dots B_K]$ consist of K cluster centers. The objective of clustering is to arrange each data object from the set S with one of the cluster center j of set P such that the value of objective function is minimized. The objective function is defined as sum of squared Euclidean distance between the data object A_i and cluster center B_j and it can be described as follows:

$$D = \sum_{i=1}^N \min_{j=1,2,3,\dots,K} \|A_i - B_j\|^2$$

where A_i denotes the i^{th} data objects, B_j denotes the j^{th} cluster center, and D denotes the distance between i^{th} data objects from the j^{th} cluster center. It is also noted that:

- Each cluster at least consists of one data object.
 $B_j \neq \emptyset$, for all $j \in \{1, 2, 3 \dots K\}$, where B_j represents the j^{th} cluster and K denotes the total number of clusters.
- Each data object is allotted to only one cluster.
 $B_i \cap B_j = \emptyset$, for all $i \neq j$ and $i, j \in \{1, 2, 3 \dots K\}$ such that i^{th} and j^{th} clusters do not consists same data objects.
- Each data objects should be allocated to a cluster.
 $\bigcup_{j=1}^K B_j = S$ where S represents the set of data objects.

Hence, the aim of the partition based clustering algorithm is to determine the K number of cluster centers in a given dataset. Here, the MCSS algorithm is applied for determining the optimal cluster centers in a dataset.

Clustering has proven its importance in many applications successfully. Some of these are pattern recognition [1, 2], image processing [3–6], process monitoring [7], machine learning [8, 9], quantitative structure activity relationship [10], document retrieval [11], bioinformatics [12], image segmentation [13], construction management [14], marketing [15, 16] and healthcare [17, 18]. Broadly, clustering algorithms can be divided into two categories - partition based clustering algorithms and hierarchal clustering algorithms. In partition based clustering algorithms, partition a dataset into k clusters on the basis of some fitness functions [19]. While in hierarchical clustering algorithms, clustering of data occurs in the form of tree representation and this representation is known as dendrogram. Hierarchical clustering algorithms do not require any prerequisite knowledge about number of clusters in a dataset but its only drawback is lacking of dynamism as the objects are tightly bound with respective clusters [20–23]. However, our research is focused on partition clustering, which decomposes the data into several disjoint clusters that are optimal in terms of some predefined criteria. From the literature, it is found that K-means algorithm is the oldest, most popular, and extensively used partition based algorithm for data clustering. It is easy, fast, simple structure, and having linear time complexity [24, 25]. In K-means algorithm, a dataset is decomposed into a predefined number of clusters and the data into distinct clusters based on the euclidean distance [25].

Nowadays, heuristic approaches gain wide popularity to solve the clustering problem and become more successful. Numerous researchers have been applying heuristic approaches in the field of clustering. Some of these are summarized as simulated annealing [26], tabu search [27, 28], genetic algorithm [29–32], particle swarm optimization [33, 34], ant colony optimization [35], artificial bee colony algorithm [36, 37, 56], charged system search algorithm [38, 39], cat swarm optimization [40–42, 57], teacher learning based optimization method [43, 44], gravitational search algorithm [45, 46] and binary search based clustering algorithm [47].

2. Magnetic charge system search (MCSS) algorithm

The magnetic charged system search (MCSS) algorithm is a recent meta-heuristic algorithm based on electromagnetic theory [48]. According to electromagnetic theory, moving charged particles produce an electric field as well as a magnetic field. Movement of the charged particles in a magnetic field enforces a magnetic force on the other charged particles and the resultant force is proportional to the charge (mass) and speed of the charged particles. The magnitude and direction of the resultant force depend on the two factors: first, velocity of the charged particles, and secondly, magnitude and direction of the magnetic field. Thus, the MCSS algorithm is further advancement in the charge system search (CSS) algorithm using the concept of electromagnetic theory. The difference between the CSS and MCSS is that the CSS algorithm considers only the electric force to determine the movement of CPs while the MCSS utilizes both the forces (electric and magnetic) to determine the same. Along this, MCSS can be either attractive or repulsive in nature. This nature of MCSS algorithm generates more promising solutions in random space. On the other hand, CSS algorithm is attractive by nature. Thus, the performance of the algorithm can be affected with small number of CPs. So, the addition of the magnetic force to the already existing electric force, results in enhancement of both the exploration and exploitation capabilities of CSS and this makes the algorithm more realistic one. Hence, the inclusion of magnetic force in the charge system search (CSS) algorithm results in the formation of a new algorithm known as magnetic charge system search (MCSS). The main steps of the MCSS algorithm are as follows.

Step 1: Initialization

Algorithm starts by identifying the initial positions of charged particles (CPs) in d-dimensional space in random order and set the initial velocities of CPs is zero. To determine the initial positions of CPs, equation 1 is used. A variable charge memory (CM) is used to store the best results.

$$C_k = X_{j_{\min}} + r_j * ((X_{j_{\max}} - X_{j_{\min}}) / K), \text{ where } j = 1, 2, \dots, d \text{ and } k = 1, 2, \dots, K \quad (1)$$

where, C_k denotes the k^{th} cluster center for a given dataset, r_j is a random number in the range of 0 and 1, $X_{j_{\min}}$ and $X_{j_{\max}}$ denote the minimum and maximum value of the j^{th} attribute of the dataset, and K represents the total number of clusters in a dataset.

Step 2: Compute the total force (F_{total}) acts on CPs.

The total force is the combination of the electric force and magnetic force, and this force influences the movement of CPs in d-dimensional space. It can be computed as follows:

- Determine the electric force – when CPs move in d-dimensional space, an electric field is produced surrounding it, and exerted an electric field on the other CPs. This electric force is directly proportional to the magnitude of its charge and the distance between CPs. The magnitude of an electric force generated by a charge particle is enforced on another charge particle and it can be measured using equation 2.

$$E_{ik} = q_k \sum_{i, i \neq k} \left(\frac{q_i}{R^3} * w_1 + \frac{q_i}{r_{ik}^2} * w_2 \right) * p_{ik} * (X_i - C_k), \begin{cases} k = 1, 2, 3, \dots, K \\ w_1 = 1, w_2 = 0 \leftrightarrow r_{ik} < R \\ w_1 = 0, w_2 = 1 \leftrightarrow r_{ik} \geq R \end{cases} \quad (2)$$

In equation 2, q_i and q_k represents the fitness values of i^{th} and k^{th} CP, r_{ik} denotes the separation distance between i^{th} and k^{th} CPs, w_1 and w_2 are the two variables whose values are either 0 or 1, R represents the radius of CPs which is set to unity and it is assumed that each CPs has uniform volume charge density but changes in every iteration, and p_{ik} denotes the moving probability of each CPs.

- Determine the magnetic force – The movement of CPs also produce magnetic field along with the electric field. As a result of this, a magnetic force is imposed on the other CPs and equation 3 is utilized to compute the magnitude of magnetic force exerted by a CP on other CPs. It can be either positive or negative depending on the value of average electric current of the previous iteration.

$$M_{ik} = q_k \sum_{i, i \neq k} \left(\frac{I_i}{R^2} * r_{ik} * w_1 + \frac{I_i}{r_{ik}} * w_2 \right) * PM_{ik} * (X_i - C_k), \begin{cases} k = 1, 2, 3, \dots, K \\ w_1 = 1, w_2 = 0 \leftrightarrow r_{ik} < R \\ w_1 = 0, w_2 = 1 \leftrightarrow r_{ik} \geq R \end{cases} \quad (3)$$

In equation 3, q_k represents the fitness values of the k^{th} CP, I_i is the average electric current, r_{ik} denotes the separation distance between i^{th} data instance and k^{th} CPs, w_1 and w_2 are the two variables whose values are either 0 or 1, R represents the radius of CPs which is set to unity, and PM_{ik} denotes the probability of magnetic influence between i^{th} data instance and k^{th} CP. In other words, it can be summarized that the magnetic force can be either attractive or repulsive in nature. As a result of this, more promising solutions can be generated during the search. Whereas, the electric force is always attractive in nature. Therefore, this nature of electric force may influence the performance of the algorithm. Hence, to overcome the repulsive nature, a probability function is added with the electric force and finally, the total force acting on other CPs can be computed using equation 4.

$$F_{\text{total}} = p_r * E_{ik} + M_{ik} \quad (4)$$

Where, p_r denotes a probability value to determine either the electric force (E_{ik}) repelling or attracting, E_{ik} and M_{ik} present the electric and magnetic forces exerted by the k^{th} CP to i^{th} data instance.

Step 3: Determine the new positions and velocities of CPs.

Newton second law of motion is applied to determine the movement of CPs. The magnitude of the total force with Newtonian laws is used to produce the next positions and velocities of CPs. The new positions and velocities of CPs can be computed using equation 5 and 6.

$$C_{k \text{ new}} = \text{rand}_1 * Z_a * \frac{F_{\text{total}}}{m_k} * \Delta t^2 + \text{rand}_2 * Z_v * V_{k \text{ old}} * \Delta t + C_{k \text{ old}} \quad (5)$$

Where, rand_1 and rand_2 are the two random variable in the range of 0 and 1, Z_a and Z_v act as control parameters to control the influence of total force (F_{total}), and $V_{k \text{ old}}$ denotes the velocity of k^{th} CPs, m_k is the mass of k^{th} CPs which is equal to the q_k , Δt represents the time step which is set to 1, and $C_{k \text{ old}}$ denotes the position of k^{th} current CP.

$$V_{k \text{ new}} = \frac{C_{k \text{ new}} - C_{k \text{ old}}}{\Delta t} \quad (6)$$

where $V_{k \text{ old}}$ denotes the new velocity of k^{th} CP, $C_{k \text{ old}}$ and $C_{k \text{ new}}$ represents the old and new position of k^{th} CP, and Δt represents the time stamp.

Step 4: Update charge memory (CM)

CPs with better objective function values replace the worst CPs from the CM and store the positions of new CPs in CM.

Step 5: Termination condition

If the maximum iterations is reached and condition is satisfied, then stop the algorithm and obtain the optimal cluster centers. Otherwise repeat steps 2–4.

2.1. Pseudo code of MCSS algorithm for clustering

This section summarizes the pseudo code of the MCSS algorithm for clustering tasks.

Step 1: Load the dataset and initialize the parameters of MCCS algorithm.

Step 2: Initialize the initial positions and velocities of Charged Particles (CPs).

Step 3: Compute the value of objective function using equation 7 and arrange the data instances to the clusters using minimum value of objective function.

$$d_{ik} = \sum_{k=1}^K \sum_{i=1}^n \sum_{j=1}^d \sqrt{X_{ji} - C_{jk}}^2 \quad (7)$$

where, X_{ji} denotes the j^{th} attribute of the i^{th} data instance, C_{jk} represents the j^{th} attribute of the k^{th} CP, and d_{ik} denotes Euclidean distance between i^{th} data instance from the k^{th} CP.

Step 4: Compute the mass of initial positioned CPs.

Step 5: Store the positions of initial CPs (C_k) into a variable, called charge memory (CM).

Step 6: While the termination conditions are not met, compute the value of Electric Force (E_{ik}) for each CPs as follows:

Step 6.1: Calculate the value of moving probability (P_{ik}) for each charged particle C_k .

Step 6.2: Compute the fitness of each instance q_i .

Step 6.3: Compute the separation distance (r_{ik}) of CPs.

Step 6.4: Compute the value of $(X_i - C_k)$.

Step 6.5: Compute the value of Electric Force (E_{ik}) for each CPs

Step 7: Determine the value of Magnetic Force (M_{ik}) for each CPs.

Step 7.1: Compute the value of average electric current (I_i).

Step 7.2: Compute the probability of magnetic influence (PM_{ik}).

Step 7.3: Compute the value of Magnetic Force (M_{ik}) for each CPs.

Step 8: Compute the total force (F_{total}) act on each CPs.

Step 9: Calculate the new positions and velocities of charged particles using equation 5 and 6.

Step 10: Recalculate the value of objective function using new positions of charge particles.

Step 11: Compare the newly generated charge particles to the charge particles reside in CM.

Step 12: Memorize the best solution achieved so far and $\text{Iteration} = \text{Iteration} + 1$;

Step 13: Output the best solution obtained.

3. Experimental results

This section deals with the experimental setup of our study. It includes the performance measures, parameters settings, datasets to be used, experiment results, and statistical analysis. To prove the effectiveness of the MCSS algorithm, 10 datasets are applied in which two datasets are artificial ones and the rest are taken from UCI repository. The proposed algorithm is

implemented in MATLAB 2010a using a computer with window operating, corei3 processor, 3.4 GHz and 4 GB RAM. Experimental outcomes of MCSS algorithm are compared with other clustering algorithms like K-means, GA [30], PSO [49], ACO [35], and CSS [38].

3.1. Performance measures

The performance of MCSS algorithm is examined over the sum of intra cluster distance and F-measure parameters. The sum of intra cluster distance can be measured in terms of best case, average case, and worst case solutions including standard deviation parameter which shows the dispersion of the data. F-measure parameter is used to measure the accuracy of proposed method. Performance measures are described as follows:

Intra cluster distances

Intra cluster distance can be used to measure the quality of clustering [35, 36]. It indicates the distance between the data objects within a cluster and its cluster center. This parameter also highlights the quality of clustering i.e. minimum is the intra cluster distance, better will be the quality of the solution. The results are measured in terms of best, average, and worst solutions.

Standard Deviation (Std.)

Standard deviation gives the information about the scattering of data within a cluster [47, 49]. Lower value of standard deviation indicates that the data objects are scattered near its center, while high value indicates that the data is dispersed away from its center point.

F-Measure

This parameter is measured in terms of recall and precision of an information retrieval system [50, 51]. It is also described in terms of weighted harmonic mean of recall and precision. Recall and precision of an information retrieval system is computed using equation 8 which can be described as follows:

$$\text{Recall}(r(i, j)) = \frac{n_{i,j}}{n_i} \text{ and Precision}(p(i, j)) = \frac{n_{i,j}}{n_j} \quad (8)$$

The value of F-measure (F (i, j)) can be computed using equation 9.

$$F(i, j) = \frac{2 * (\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})} \quad (9)$$

Finally, the value of F-measure for a given clustering algorithm which consists of n number of data instances is calculated using equation 10.

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

(10)

3.2. Parameters settings

In order to evaluate the performance of the proposed algorithm, user defined parameters are to be used prior to the process. In MCSS, there are four user defined parameters such as number of CPs, rand, R and ϵ . The details of the parameters as follows: the number of CPs is equal number of clusters present in a dataset, rand is a random function that provides a value in the range of 0 and 1, R denotes the radius of CPs and it is set as 1, ϵ is also a user defined parameter which is used to prevent the singularity and it is set to 0.001. In addition to it, number of iterations for algorithm must be specified. Therefore, maximum iteration number is set to 100 and results are summed over 10 runs of the algorithm using different initial cluster centers for each dataset. **Table 1** summarizes the parameters setting of MCSS algorithm. It is also mentioned that the performance of the proposed algorithm is compared with the K-means, GA, PSO, ACO, and CSS. The parameter settings of these algorithms are set accordingly as reported (**Figures 1 and 2; Table 2**).

Parameters	Value
No. of CPs	No. of Clusters
rand	random value between [0, 1]
R	1
ϵ	0.001

Table 1. Parameters setting of MCSS algorithm.

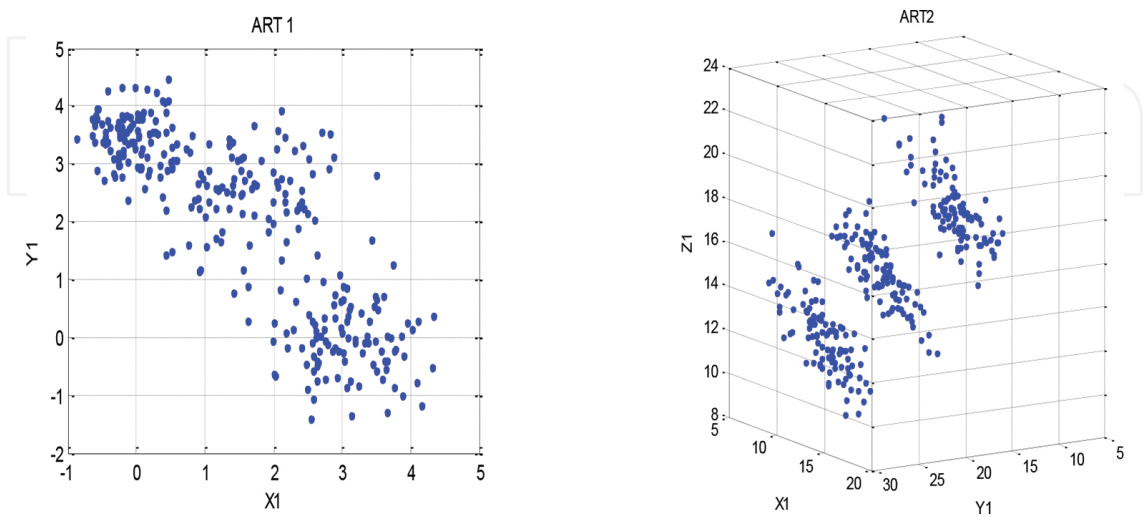


Figure 1. (a): Distribution of data in ART1. (b): Distribution of data in ART2.

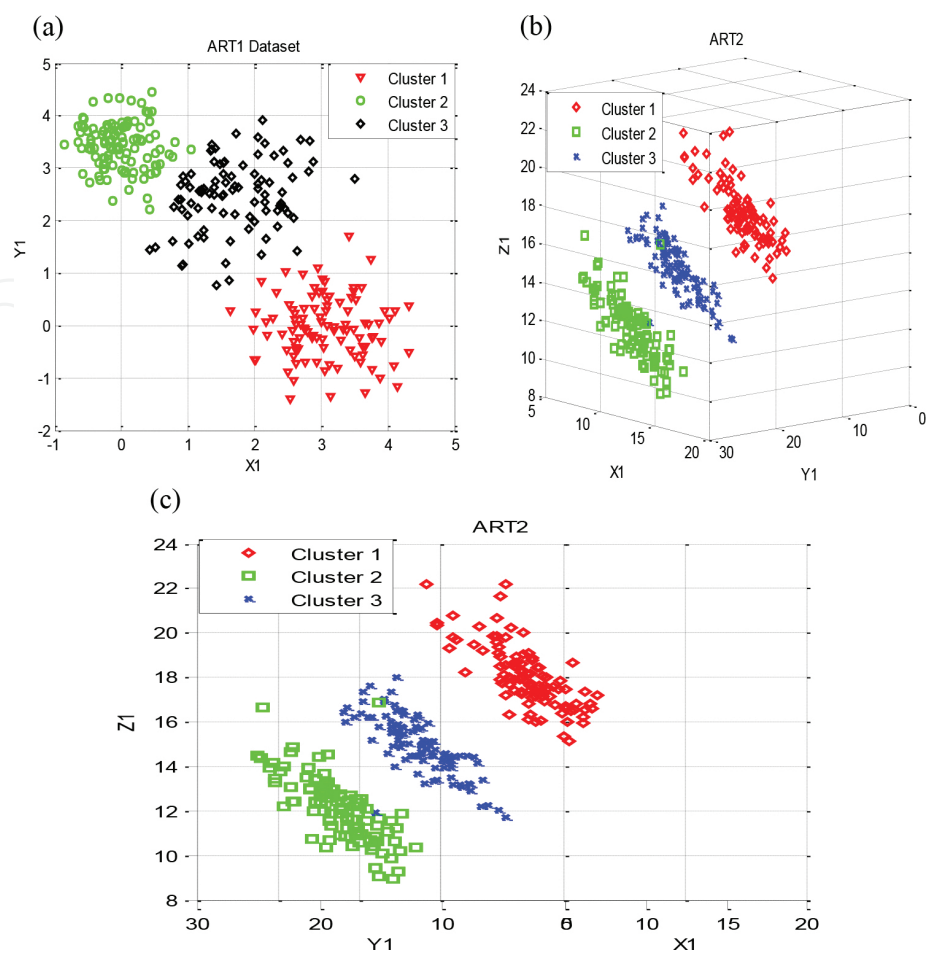


Figure 2. (a): Clustering in ART1 dataset. (b): Clustering the ART2 dataset. (c): Clustering the ART1 dataset using MCSS (Vertical view as X1 and Y1 coordinate in horizontal plane and Z1 coordinate in vertical plane).

Dataset	Classes	Attributes	Total instances	Instance in each classes
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 2. Description of datasets.

3.3. Experiment results

This subsection demonstrates the results of the proposed algorithm. The results of the proposed algorithm are compared with other existing techniques like K-means, GA, PSO, ACO, and CSS using a mixture of datasets [53–55]. Two artificial and eight real life datasets are used to obtain the results [52]. In real life datasets, iris, thyroid, and vowel datasets are categorized as low dimensional datasets, while cancer and LD datasets are moderate ones, and the rest of datasets (wine, CMC, and glass) are high dimensional. For enrich visualization and understanding, the results are discussed with one dataset at a time.

Dataset	Parameters	K-means	GA	PSO	ACO	CSS	MCSS
ART 1	Best	157.12	154.46	154.06	154.37	153.91	153.18
	Average	161.12	158.87	158.24	158.52	158.29	158.02
	Worst	166.08	164.08	161.83	162.52	161.32	159.26
	Std	0.34	0.281	0	0	0	0
	F-Measure	99.14	99.78	100	100	100	100

Table 3. Comparison of the proposed MCSS algorithm with other clustering algorithms using ART1 dataset.

Table 3 illustrates the results of the proposed method as well as other clustering algorithms (in terms of intra cluster distance: best, average and worst, standard deviation, and F-measure parameters) for ART1 dataset. It is seen that the K-means exhibits the poor performance among all the techniques being compared using all of the parameters. From the results, it also noticed that performance of the PSO, ACO, CSS, and MCSS are almost similar except intra cluster parameter. On the behalf of intra cluster parameter, it can be said that the MCSS algorithm achieves minimum distance in comparison to all other algorithms.

Dataset	Parameters	K-means	GA	PSO	ACO	CSS	MCSS
ART2	Best	743	741.71	740.29	739.81	738.96	737.85
	Average	749.83	747.67	745.78	746.01	745.61	745.12
	Worst	754.28	753.93	749.52	749.97	749.66	748.67
	Std	0.516	0.356	0.237	0.206	0.209	0.17
	F-Measure	98.94	99.17	99.26	99.19	99.43	99.56

Table 4. Comparison of the proposed MCSS algorithm with other clustering algorithms using ART2 dataset.

Table 4 summarizes the results of all the techniques for artificial dataset ART2. From the results, it is clearly shown that a significant difference occurred between the results of the proposed algorithm and other algorithms. The proposed algorithm outperforms using all of the parameters. Again, it is observed that the performance of the K-means algorithm is poor

among all the methods. It is also stated that the results of the CSS and PSO algorithms are close to the optimal solution, but with slightly high value of standard deviation parameter.

Table 5 displays the results of the proposed algorithm and other algorithms for iris dataset. From this table, it came to notice that results obtained using GA is far from the optimal solutions, while the proposed method again gives the superior results. It is also observed that K-means algorithm gives the better results with iris dataset; especially it performs well over GA and ACO algorithms. It is also noticed that results of the K-means algorithm is very close to the PSO algorithm (in terms of F-measure parameter).

Results of all six methods for wine dataset are listed in **Table 6**. It demonstrates that MCSS algorithm obtains good results (in terms of intra cluster distance and F-measure) in comparison to others, but slightly large value of standard deviation parameter. On the other hand, it is also stated that the performance of GA, PSO, and ACO are nearly same except some variation between F-measure parameter. Again, the performance of the K-means algorithm is better than the GA, PSO, and ACO in terms of F-measure parameter but with a large value of standard deviation parameter. CSS algorithm also gives good performance with wine dataset except MCSS, and obtains low value for standard deviation parameter which shows that in each iteration, a near optimal solution is generated.

Dataset	Parameters	K-means	GA	PSO	ACO	CSS	MCSS
Iris	Best	97.33	113.98	96.89	97.1	96.47	96.34
	Average	106.05	125.19	97.23	97.17	96.63	96.57
	Worst	120.45	139.77	97.89	97.8	96.78	96.63
	Std	14.631	14.563	0.347	0.367	0.14	0.1
	F-Measure	0.782	0.778	0.782	0.779	0.787	0.790

Table 5. Comparison of the proposed MCSS algorithm with other clustering algorithms using iris dataset.

Dataset	Parameters	K-means	GA	PSO	ACO	CSS	MCSS
Wine	Best	16555.68	16530.53	16345.96	16530.53	16282.12	16158.56
	Average	18061	16530.53	16417.47	16530.53	16289.42	16189.96
	Worst	18563.12	16530.53	16562.31	16530.53	16317.67	16223.61
	Std	793.213	0	85.497	0	10.31	36.72
	F-Measure	0.521	0.515	0.518	0.519	0.529	0.537

Table 6. Comparison of the proposed MCSS algorithm with other clustering algorithms using wine dataset.

Table 7 describes the results of all the six algorithms using LD dataset. From the results, it is clearly seen that the outcomes of the proposed algorithm is better in comparison to the other algorithms. It is also noted that K-means algorithm does not perform well with LD dataset, and results obtained using K-means are far-far away from the optimal ones. Again, it came

into revelation that the performance of PSO algorithm is better except MCSS algorithm and its results are close to the optimal solutions.

Dataset	Parameters	K-means	GA	PSO	ACO	CSS	MCSS
LD	Best	11397.83	532.48	209.15	224.76	207.09	206.14
	Average	11673.12	543.69	224.47	235.16	228.27	221.69
	Worst	12043.12	563.26	239.11	256.44	242.14	236.23
	Std	667.56	41.78	29.38	17.46	18.54	12.07
	F-Measure	0.467	0.482	0.493	0.487	0.491	0.495

Table 7. Comparison of the proposed MCSS algorithm with other clustering algorithms using LD dataset.

Dataset	Parameters	K-means	GA	PSO	ACO	CSS	MCSS
Cancer	Best	2999.19	2999.32	2973.5	2970.49	2946.48	2932.43
	Average	3251.21	3249.46	3050.04	3046.06	2961.16	2947.74
	Worst	3521.59	3427.43	3318.88	3242.01	3006.14	2961.03
	Std	251.14	229.734	110.801	90.5	12.23	10.33
	F-Measure	0.829	0.819	0.819	0.821	0.847	0.859

Table 8. Comparison of the proposed MCSS algorithm with other clustering algorithms using cancer dataset.

Results of all the six algorithm for cancer dataset is listed in **Table 8**. As it indicates, the performance of the GA and PSO algorithms are not so good with cancer dataset. But the proposed algorithm works well and achieves respectable results as compared to others. K-means algorithm also achieves good results over GA, ACO, and PSO algorithms.

Dataset	Parameters	K-means	GA	PSO	ACO	CSS	MCSS
CMC	Best	5842.2	5705.63	5700.98	5701.92	5672.46	5653.26
	Average	5893.6	5756.59	5820.96	5819.13	5687.82	5678.83
	Worst	5934.43	5812.64	5923.24	5912.43	5723.63	5697.12
	Std	47.16	50.369	46.959	45.634	21.43	17.37
	F-Measure	0.334	0.324	0.331	0.328	0.359	0.368

Table 9. Comparison of the proposed MCSS algorithm with other clustering algorithms using CMC dataset.

Table 9 demonstrates the results of all the six algorithms for CMC dataset. As can be seen clearly, the proposed method achieves better results in comparison to the other algorithms using all the parameters. It is also stated that the performance of the GA is found to be poor among all the algorithms (in terms of standard deviation and F-measure parameters). Along

this, it is found that the K-means algorithm obtains maximum intra cluster distance amongst all.

Table 10 illustrates the results of the proposed algorithm and all other algorithms for thyroid dataset. From the results, again it is observed that proposed algorithm obtains superior results in comparison to other algorithms, but gets the marginally high value of standard deviation parameter in comparison to GA, PSO, and ACO algorithms. Again, K-means results are far away from the optimal ones and CSS algorithm also obtain high value of standard deviation parameter except K-means. In the rest of the algorithms, performance of ACO algorithm is better.

Dataset	Parameters	K-means	GA	PSO	ACO	CSS	MCSS
Thyroid	Best	13956.83	10176.29	10108.56	10085.82	9997.25	9928.89
	Average	14 133.14	10218.82	10149.7	10108.13	10078.23	10036.93
	Worst	146424.21	10254.39	10172.86	10134.82	10116.52	10078.34
	Std	246.06	32.64	27.13	21.34	49.02	43.61
	F-Measure	0.731	0.763	0.778	0.783	0.789	0.793

Table 10. Comparison of the proposed MCSS algorithm with other clustering algorithms using thyroid dataset.

Results of all the six algorithm for glass dataset is summarized in **Table 11**. It indicates that CSS algorithm gives the better results for glass dataset in comparison to the other algorithms (in terms of intra cluster distance and standard deviation parameters). From the analysis of F-measure parameter, it is found that the performance of the MCSS is better than CSS. It is worthy to be noted that both the CSS and MCSS achieve good results on the cost of standard deviation parameter. Along this, it is also noticed that the GA exhibits weak performance.

Dataset	Parameters	K-means	GA	PSO	ACO	CSS	MCSS
Glass	Best	215.74	278.37	270.57	269.72	203.58	209.47
	Average	235.5	282.32	275.71	273.46	223.44	231.61
	Worst	255.38	286.77	283.52	280.08	241.27	263.44
	Std	12.47	4.138	4.55	3.584	13.29	17.08
	F-Measure	0.431	0.333	0.359	0.364	0.446	0.449

Table 11. Comparison of the proposed MCSS algorithm with other clustering algorithms using glass dataset.

Table 12 summarizes the results of the proposed algorithm and all other algorithms for vowel dataset. From the results, it is noticed that MCSS algorithm obtains minimum intra

cluster distance amongst all but on the cost of high value of standard deviation parameter. In addition to it, it is also observed that both the MCSS and K-means algorithms exhibit similar performance in terms of F-measure parameters, but K-means obtains minimum value for standard deviation parameter. It is also stated that the quality of clustering is measured in terms of intra cluster distance. Therefore, MCSS algorithm provides good quality results in terms of intra cluster distance. Whereas, ACO method gets maximum intra cluster distance among all the methods. Over all, it is concluded that the proposed algorithm gives better performance with most of datasets in comparison to the other algorithms and quality of solutions is obtained. A statistical analysis is also carried out to prove the same.

Dataset	Parameters	K-means	GA	PSO	ACO	CSS	MCSS
Vowel	Best	149422.26	149513.73	148976.01	149395.6	149335.61	146124.87
	Average	159242.89	159153.49	151999.82	159458.14	152128.19	149832.13
	Worst	161236.81	165991.65	158121.18	165939.82	154537.08	157726.43
	Std	916	3105.544	2881.346	3485.381	2128.02	2516.58
	F-Measure	0.652	0.647	0.648	0.649	0.649	0.652

Table 12. Comparison of the proposed MCSS algorithm with other clustering algorithms using vowel dataset.

4. Conclusion

In this chapter, a magnetic charged system search algorithm is applied to solve the clustering problems. The idea of proposed algorithm came from the electromagnetic theory and it is based on the behavior of moving charged particles. A moving charged particle exerts both the forces (electric force and magnetic force) on other charged particles and in turn altered the positions of charged particles. Therefore, in MCSS algorithm, initial population is presented in the form of charged particles. It utilizes the concept of electric and magnetic forces along with newton second law of motion to obtain the updated positions of charged particles. In MCSS, both the electric force (E_k) and magnetic force (M_k) correspond to the local search for the solution, while the global solution is exploited using newton second law of motion. The aim of this research is to investigate the applicability of MCSS algorithm for clustering problems. To achieve the same, performance of the MCSS algorithm is evaluated on variety of datasets and compared with K-Means, GA, PSO, ACO, and CSS using intra cluster distance, standard deviation, and F-measure parameters. Experiment results support the applicability of proposed algorithm in clustering field as well as the proposed method provides good results with most datasets in comparison to the other methods. Finally, it is concluded that proposed method not only gives good results but also improves the quality of solutions.

Author details

Yugal Kumar^{1*}, Sumit Gupta^{1*}, Dharmender Kumar^{2*} and Gadadhar Sahoo^{3*}

*Address all correspondence to: yugalkumar.14@gmail.com; sumitkumarbsr19@gmail.com; dharm_india@yahoo.com; gsahoo@bistmesra.ac.in

1 Department of Information Technology, Krishna Institute of Engineering and Technology, Ghaziabad, India

2 Department of Computer Science and Engineering, GJU, Hissar, India

3 Department of Computer Science and Engineering, Birla Institute of Technology, Mesra, Ranchi, India

References

- [1] Webb, A. Statistical pattern recognition. New Jersey: John Wiley & Sons, pp. 361–406 (2002).
- [2] Zhou, H., and Liu, Y. Accurate integration of multi-view range images using k-means clustering. *Pattern Recognition*, 41, no. 1 (2008), 152–175.
- [3] Sonka, M., Hlavac, V., and Boyle, R. Image processing, analysis, and machine vision. Springer, US, 1993. (1999).
- [4] Das, S., and Konar, A. Automatic image pixel clustering with an improved differential evolution. *Applied Soft Computing*, 9, no. 1 (2009), pp. 226–236.
- [5] Portela, N. M., Cavalcanti, G. D., and Ren, T. I. Semi-supervised clustering for MR brain image segmentation. *Expert Systems with Applications*, 41, no. 4 (2014), pp. 1492–1497.
- [6] Siang Tan, K., and Mat Isa, N. A. Color image segmentation using histogram thresholding –Fuzzy C-means hybrid approach. *Pattern Recognition*, 44, no. 1 (2011), pp. 1–15.
- [7] Teppola, P., Mujunen, S.-P. and Minkkinen, P. Adaptive Fuzzy C-Means clustering in process monitoring. *Chemometrics and Intelligent Laboratory Systems*, 45, no. (1999), pp. 23–38.
- [8] Alpaydin, E. Introduction to machine learning. Cambridge, Massachusetts, London, England. MIT Press, 2004.
- [9] Anaya, A. R., and Boticario, J. G. Application of machine learning techniques to analyse student interactions and improve the collaboration process. *Expert Systems with Applications*, 38, no. 2 (2011), pp. 1171–1181.

- [10] Dunn III, W. J., Greenberg, M. J. and Callejas, S. S. Use of cluster analysis in the development of structure-activity relations for antitumor triazenes. *Journal of Medicinal Chemistry*, 19, no. 11 (1976), pp. 1299–1301.
- [11] Hu, G., Zhou, S., Guan, J. and Hu, X. Towards effective document clustering: a constrained K -means based approach. *Information Processing & Management*, 44, no. 4 (2008), pp. 1397–1409.
- [12] He, Y., Pan, W., and Lin, J. Cluster analysis using multivariate normal mixture models to detect differential gene expression with microarray data. *Computational Statistics & Data Analysis*, 51, no. 2 (2006), pp. 641–658.
- [13] Pappas, Thrasyvoulos N., An adaptive clustering algorithm for image segmentation. *IEEE Transactions on Signal Processing*, 40, no. 4 (1992), pp. 901–914.
- [14] Cheng, Y.M., and Leu, S. S., Constraint-based clustering and its applications in construction management. *Expert Systems with Applications*, 36 (2009), 5761–5767.
- [15] Kim, K.J. and Ahn, H. A recommender system using {GA} k-means clustering in an online shopping market. *Expert Systems with Applications*, 34 (2008), pp. 1200–1209.
- [16] Kuo, R., An, Y., Wang, H., and Chung, W., Integration of self-organizing feature maps neural network and genetic k-means algorithm for market segmentation. *Expert Systems with Applications*, 30 (2006), pp. 313–324.
- [17] Gunes, S., Polat, K., and Sebnem, Y. Efficient sleep stage recognition system based on {EEG} signal using k-means clustering based feature weighting. *Expert Systems with Applications*, 37 (2010), pp. 7922–7928.
- [18] Hung, Y.S., Chen, K.L. B., Yang, C.T., and Deng, G.F. Web usage mining for analysing elder self-care behavior patterns. *Expert Systems with Applications*, 40 (2013), pp. 775–783.
- [19] Jain, A.K. Data clustering: 50 years beyond K-means. *Pattern Recognition Letter*, 31 (2010), pp. 651–666.
- [20] Barbakh, W., Wu, Y., Fyfe, C. Review of Clustering Algorithms. Non-Standard Parameter Adaptation for Exploratory Data Analysis. Springer, Berlin/Heidelberg (2009), pp. 7–28.
- [21] Camastra, F., Vinciarelli, A. Clustering Methods. *Machine Learning for Audio, Image and Video Analysis*. Springer, London (2008), pp. 117–148.
- [22] Kogan, J., Nicholas, C., Teboulle, M., Berkhin, P. A Survey of Clustering Data Mining Techniques, Grouping Multidimensional Data. Springer, Berlin Heidelberg (2006), pp. 25–71.
- [23] Maimon, O., Rokach, L., A survey of Clustering Algorithms, *Data Mining and Knowledge Discovery Handbook*. Springer, US (2010), pp. 269–298.

- [24] MacQueen, J. Some methods for classification and analysis of multivariate observations. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. 1. no. 14 (1967) pp. 281–297.
- [25] Jain, A.K., Murty, M.N., Flynn, P.J. Data clustering: a review. ACM Computing Surveys (1999) Vol. 31, Issue No. 3, pp. 264–323.
- [26] Selim, S.Z., Alsultan, K. A simulated annealing algorithm for the clustering problem. Pattern Recognition 24 (1991), pp. 1003–1008.
- [27] Al-Sultan, K.S. A Tabu search approach to the clustering problem. Pattern Recognition 28 (1995), pp. 1443–1451.
- [28] Sung, C.S. and Jin, H.W. A tabu-search-based heuristic for clustering. Pattern Recognition 33 (2000), 849–858.
- [29] Cowgill, M.C., Harvey, R.J., and Watson, L.T. A genetic algorithm approach to cluster analysis. Comput. Math. Appl. 37 (1999), pp. 99–108.
- [30] Krishna, K., Narasimha Murty, M. Genetic K-means algorithm. IEEE Transactions on Systems, Man, Cybernet. Part B: Cybernet. 29 (1999), pp. 433–439.
- [31] Maulik, U., Bandyopadhyay, S. Genetic algorithm-based clustering technique. Pattern Recognition 33 (2000), pp. 1455–1465.
- [32] Murthy, C.A., Chowdhury, N. In search of optimal clusters using genetic algorithms. Pattern Recognition Lett. 17 (1996), pp. 825–832.
- [33] Van der Merwe, D. W. and Engelbrecht, A. P. Data clustering using particle swarm optimization, In Congress on Evolutionary Computation CEC-03, 1, pp. 215–220. IEEE (2003).
- [34] Rana, S., Jasola, S. and Kumar, R. A review on particle swarm optimization algorithms and their applications to data clustering. Artificial Intelligence Review 35, no. 3 (2011) pp. 211–222.
- [35] Shelokar, P. S., Jayaraman, V. K. and Kulkarni, B. D. An ant colony approach for clustering. Analytica Chimica Acta 509, no. 2 (2004), pp. 187–195.
- [36] Zhang, C., Ouyang, D. and Ning, J. An artificial bee colony approach for clustering. Expert Systems with Applications 37, no. 7 (2010), pp. 4761–4767.
- [37] Karaboga, D., and Ozturk, C. A novel clustering approach: artificial bee colony (ABC) algorithm. Applied Soft Computing 11, no. 1 (2011), 652–657.
- [38] Kumar, Y., and Sahoo, G. A charged system search approach for data clustering. Progress in Artificial Intelligence 2, no. 2–3 (2014), pp. 153–166.
- [39] Kumar, Y. and Sahoo, G. A chaotic charged system search approach for data clustering. Informatica, 38, no. 3 (2014), pp. 249–261.

- [40] Santosa, B., and Ningrum, M. K. Cat swarm optimization for clustering. In *Soft Computing and Pattern Recognition, 2009. SOCPAR'09. International Conference of*, pp. 54–59. IEEE (2009).
- [41] Kumar, Y. and Sahoo, G. A Hybridize approach for data clustering based on cat swarm optimization. *International Journal of Information and Communication Technology* (2015) (accepted for publication).
- [42] Kumar, Y. and Sahoo, G. An improved cat swarm optimization algorithm for data clustering. In *International Conference on Computational Intelligence in Data Mining (ICCIDM-2014) proceedings published in the Springer's Series on Smart Innovation, Systems and Technologies, Germany* (2014).
- [43] Satapathy, S. C. and Naik, A. Data clustering based on teaching-learning-based optimization. In *Swarm, Evolutionary, and Memetic Computing*, pp. 148–156. Springer Berlin Heidelberg (2011).
- [44] Sahoo, A. J. and Kumar, Y. Modified teacher learning based optimization method for data clustering. In *Advances in Signal Processing and Intelligent Recognition Systems* (2014), pp. 429–437.
- [45] Hatamlou, A., Abdullah, S., Nezamabadi-pour, H. Application of Gravitational Search Algorithm on Data Clustering, *Rough Sets and Knowledge Technology*. Springer, Berlin/Heidelberg (2011), pp. 337–346.
- [46] Hatamlou, A., Abdullah, S., Nezamabadi-pour, H. A combined approach for clustering based on K-means and gravitational search algorithms. *Swarm and Evolutionary Computation*, 6 (2012), pp. 47–52.
- [47] Hatamlou A. In search of optimal centroids on data clustering using a binary search algorithm. *Pattern Recognition Letters* 33 (2012), pp. 1756–1760.
- [48] Kaveh, A., Motie Share, M. A. and Moslehi, M. Magnetic charged system search: a new meta-heuristic algorithm for optimization. *Acta Mechanica* 224, no. 1 (2013), pp. 85–107.
- [49] Niknam, T. and Amiri, B. An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis. *Applied Soft Computing* 10 (2010), pp. 183–197.
- [50] Dalli, A. Adaptation of the F-measure to cluster based lexicon quality evaluation. In *Proceedings of the EACL* (2003), pp. 51–56.
- [51] Handl, J., Knowles, J., and Dorigo, M. On the performance of ant-based clustering. *Design and Application of Hybrid Intelligent System. Frontiers in Artificial Intelligence and Applications*, 104 (2003), pp. 204–213.
- [52] <https://archive.ics.uci.edu/ml/datasets>
- [53] Demšar, J. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research* 7 (2006), pp. 1–30.

- [54] Derrac, J. Salvador García, Daniel Molina, and Francisco Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation* 1, no. 1 (2011), pp. 3–18.
- [55] García, S., Fernández, A., Luengo, J., and Herrera, F. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. *Information Sciences* 180, no. 10 (2010), pp. 2044–2064.
- [56] Kumar, Y. and Sahoo, G. A two-step artificial bee colony algorithm for clustering. *Neural Computing and Applications*. (2015), pp 1–15, DOI: 10.1007/s00521-015-2095-5.
- [57] Kumar, Y. and Sahoo, G.. A hybrid data clustering approach based on improved cat swarm optimization and K-harmonic mean algorithm. *AI Communications (Preprint)* (2015), pp.1-14.

