HYBRIDIZATION OF MAGNETIC CHARGE SYSTEM SEARCH METHOD FOR EFFICIENT DATA CLUSTERING

Yugal Kumar[1] and G. Sahoo[2]

[1] Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat, Himachal Pardesh, India

[2] Department of Computer Science and Engineering, Birla Institute of Technology, Mesra, Ranchi, Jharkhand, India

Email: yugalkumar.14@gmail.com[1], gsahoo@bitmesra.ac.in[2]

DOI: https://doi.org/10.22452/mjcs.vol31no2.2

ABSTRACT

MCSS is a relatively new meta-heuristic algorithm inspired from the electromagnetic theory and has shown better potential than the same class of algorithms. But, like the other meta-heuristic algorithm, some performance issues are also associated with this algorithm such as convergence rate and trap in local optima. So, in this work, an attempt is made to improve the convergence rate of MCSS algorithm and proposed a Hybrid Magnetic Charge System Search (HMCSS) for solving the clustering problems. Further, a local search strategy is also inculcated into MCSS algorithm to reduce the probability of trapping in local optima and exploring promise solutions. The effectiveness of the proposed algorithm is tested on some benchmark functions and also applied to solve real world clustering problems. The experimental results show that the proposed algorithm gives better results than the existing algorithms, and also improves the convergence rate of MCSS algorithm.

Keywords: ***charge particles, clustering, electric force, magnetic force***

1.0 INTRODUCTION

From past few decades, clustering problems are attracting a bulk of attention from researchers to find the solution both theoretical and practical point of view. It is an unsupervised classification technique, allocate the data into distinct groups, where these groups are known as clusters and data within a cluster having more common characteristics. Hence, the aim of clustering is to find the optimal cluster centers of the data so that the unseen patterns can be accessed from [1-3]. These problems have been surfaced out from a variety of research fields such as pattern recognition, data analysis, image processing, market research, process monitoring, bioinformatics, biology, and so on [4-10]. Therefore, it is a major issue to either ameliorate the present problem or to develop new clustering algorithms for enhancing the efficiency of data clustering. Several clustering algorithms have been reported by research community in recent years. These algorithms can be categorized as partitioned based clustering algorithm, hierarchical clustering algorithm, density based clustering algorithm, grid based clustering algorithm, and model based clustering algorithms [11-15]. This work is primarily focused on the partition based clustering problems. In partition based clustering algorithms, different partitions of dataset are formed using a criterion function and usually Euclidean distance is used for this. K-Mean is one of the oldest, simplest and popular one partition based clustering algorithms, but it converges in local optima and its performance also depends on initial cluster centers [16-17]. To get rid of this local optima problem, many evolutionary and swarm based approaches have been applied to date. These approaches consist of innovative and intelligent paradigm to solve the optimization problems. Some of these are genetic algorithms (GA) [18, 19], particle swarm optimization (PSO) [20,21], ant colony optimization (ACO) [22, 23], artificial bee colony optimization (ABC) [24, 25], teacher learning based optimization method (TLBO) [26, 27], charge system search (CSS) [28, 29], simulated annealing (SA) [30, 31], tabu search (TS) [32], cat swarm optimization (CSO) [33-36], and magnetic charge system search (MCSS) [37-38], which have been previously implemented successfully to solve the clustering problems. These algorithms explore

the solution space for optimal solution effectively. Further, in literature, hybrid variants of these algorithms are also reported [21, 23, 25, 34, 36].

In this work, a recently developed meta-heuristic, called Magnetic charge system search (MCSS) algorithm is explored to solve the partition based clustering problems. MCSS has been used to solve engineering design problems [39]. This algorithm is inspired from the electromagnetic forces (electrical and magnetic forces between the charged particles) from physics and law of motion from mechanics. Charged particles work like search agents and explore the search space to obtain the optimal solution. Each charged particle is directed towards its global optimum position using the net amount of forces (electrical and magnetic forces) and law of motion. Nevertheless, like many other meta-heuristic and evolutionary algorithms, the MCSS algorithm also encounters some challenges. Although, MCSS is a well efficient optimization algorithm there is some inefficiency in context with the solution search equation, which in turn is used to generate new candidate solutions based on the information of previous solutions and electromagnetic forces. Hence, it is observed that the search equation of MCSS algorithm is substantial for exploitation but poor for exploration. As a result of this, it cannot effectively use information to compute the most promising search direction and due to this, issues in relation with the trapping in local optima as well effect on convergence speed may arise. To handle the aforementioned issues, an improved search mechanism is proposed for MCSS algorithm, inspired from the DE algorithm and also to guide the search of new candidate solutions in order to improve convergence rate. Further, a local search strategy is proposed to explore the more promising solution space and also to escape from the local optima problem. The boundary level constraints are also handled by using this strategy. The rest of the paper is organized as follows. Sections 2.0 and 2.1 summarize background on clustering and magnetic charged system search algorithm; Section 3.0 describes proposed HMCSS algorithm in detail; Experimental results are illustrated in section 4.0 and work is concluded in section 5.0.

2.0 BACKGROUND

In the field of data analysis, clustering is a common problem and it becomes NP hard problem when number of clusters is more than three. Consider, a dataset D(x, z), where x represents number of attributes and z represents number of clusters. Mathematically, it can be expressed as:

$$\mathrm{x} = \{\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3 \dots \dots \mathrm{x}_\mathrm{m}\} \text{ where } \mathrm{x}_\mathrm{i} \in \mathrm{R}^\mathrm{N}, \text{ and}$$
$$\mathrm{z} = \{\mathrm{z}_1, \mathrm{z}_2, z_3 \dots \dots \mathrm{z}_\mathrm{k}\} \text{ where } \forall_{\mathrm{i}\neq\mathrm{j}}\mathrm{z}_\mathrm{i} \cap \mathrm{z}_\mathrm{j} = \emptyset, \quad \bigcup_{\mathrm{i}=1}^{\mathrm{k}} \mathrm{z}_\mathrm{i} = \mathrm{x}, \quad \forall_\mathrm{i}\mathrm{z}_\mathrm{i} = \emptyset \qquad (1)$$

The goal of clustering problem is to determine the partitions in a dataset with minimum criterion function. Elements within a partition are generally homogenous in nature but exhibit heterogeneity with the elements of other partitions. Commonly, Euclidean distance is used as criterion function. It can be described as follows.

$$\mathrm{F}(\mathrm{A}, \mathrm{B}) = \sum_{\mathrm{i}=1}^{\mathrm{N}} \min \left\| \mathrm{A}_\mathrm{i} - \mathrm{B}_\mathrm{j} \right\|^2, \qquad \mathrm{j} = 1, 2, 3, \dots \dots . \mathrm{K} \qquad (2)$$

Where, A_i denotes the ith data objects, B_j denotes the jth cluster center and D denotes the distance between ith data objects from the jth cluster center. Hence, aim of the partition based clustering algorithm is to compute the optimal cluster centers in a given dataset.

2.1Magnetic charge system search

Magnetic charge system search (MCSS) is a recent meta-heuristic algorithm reported in [39]. It is based on the electromagnetic forces and newton law of motion. The candidate solution represents in terms of charged particles (CPs) and CPs are responsible for searching the optimal solution by exploring the solution space. When, a charge particle (CP) is placed in solution space, it generates its own electric field and imposes an electric force on other CPs. The amount of electric force generated by CP is calculated using equation 3.

$$E_k = q_k \sum_{i,i\neq k} \left(\frac{q_i}{R^3} * w_1 + \frac{q_i}{r_{ki}^2} * w_2 \right) * p_{ki} * (X_i - X_k), \qquad \begin{cases} k = 1, 2, 3, \ldots . K \\ w_1 = 1, w_2 = 0 \leftrightarrow r_{ik} < R \\ w_1 = 0, w_2 = 1 \leftrightarrow r_{ik} \geq R \end{cases} \tag{3}$$

where, qi and q_k represents the fitness values of i^{th} and k^{th} CP, $r_{i,k}$ denotes the separation distance between i^{th} and k^{th} CPs, w1 and w2 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. Now, the value of electric force (E_{ik}) depends on the variables q_i, q_k, $r_{i,k}$, P_{ik} and R.
Whereas, movement of CP in random space search produces the magnetic field, which in turn imposes the magnetic force on other CPs and it can be determined using equation 4.

$$M_k = q_k \sum_{i,i\neq k} \left(\frac{I_i}{R^2} * r_{ki} * w_1 + \frac{I_i}{r_{ki}} * w_2 \right) * PM_{ki} * (X_i - X_k), \begin{cases} k = 1, 2, 3, \ldots . K \\ w_1 = 1, w_2 = 0 \leftrightarrow r_{ik} < R \\ w_1 = 0, w_2 = 1 \leftrightarrow r_{ik} \geq R \end{cases} \tag{4}$$

where, q_k represents the fitness values of the k^{th} CP, Ii is the average electric current, $r_{i,k}$ 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 PM_{ik} denotes the probability of magnetic influence. Now, the value of magnetic force (M_{ik}) depends on the variables q_k, I_i, $r_{i,k}$, PM_{ik} and R.

Finally, the total force (F_{total}) acting on a CP is the synergistic action of both the electric and magnetic forces and can be computed using equation5.

$$F_{total} = p_r * E_k + M_k \tag{5}$$

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} CPs to i^{th} particles.
The total force with Newton second law of motion is used to find the updated positions of CPs in D-dimensional search space and these updated positions of CPs can be computed using the equation 6 given below.

$$X_{k,new} = rand_1 * Z_a * \frac{F_{total}}{m_k} * \Delta t^2 + rand_2 * Z_v * V_{kold} * \Delta t + X_{kold} \tag{6}$$

where, rand1 and rand2 are the two random variable in between 0 and 1, Z_a and Z_v act as control parameters to control the influence of total force (F_{total}) 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,$X_{k\,old}$ represents the position of old CPs, $V_{k\,old}$ denotes the velocity of k^{th} CPs. The updated velocities of CPs can be calculated using equation 7.

$$V_{knew} = \frac{X_{knew} - X_{kold}}{\Delta t} \tag{7}$$

3.0 PROPOSED HMCSS

This section describes the proposed HMCSS algorithm for improving the convergence rate and local optima problems of MCSS algorithm. In order to make the MCSS algorithm more efficient, robust and effective, two amendments are proposed. For achieving the good convergence rate, both the exploration and exploitation abilities of algorithm should be well balanced. MCSS algorithm is good for exploration, but converges slowly due to lack of exploitation. Another issues regarding performance of MCSS algorithm is local optima problem and boundary constraints. The proposed

amendments are explained in subsections 3.1 and 3.2, and the pseudo code of proposed HMCSS is described in subsection 3.3.

3.1 Solution Search Mechanism

The exploration and exploitation are prerequisites of each meta-heuristic algorithm. Exploration refers to the ability of the algorithm in finding global optimum solution, whereas, exploitation refers to the ability of the algorithm in finding better solution based on the previous solution [40]. In MCSS algorithm, the new candidate solution is generated using equation 6. In equation 6, the right hand side first term is responsible for the exploitation of solutions in search space and liability of exploration belongs to the second term. So, the updated solution is the combination of the total force exerted by CPs and its velocities with the old position of CPs in solution space. On the other hand, in equation 6, $rand_1$ and $rand_2$ are the two random numbers in the range of 0 and 1; Z_a and Z_v are the control parameters to control the exploitation and exploration process; $F_{total,k}$ and m_k denotes the acting force and mass of the k^{th} CP; $V_{k,old}$ is the previous velocity of k^{th} CP and $C_{k,old}$ is the position of the k^{th} CP in solution space. The equation 6 is good for exploration but poor at exploitation. It is noted that well-made search equations improve the performance of algorithm. Hence, a new search equation is proposed to enhance the exploration ability of the MCSS algorithm using the mutation strategy of Differential evolution (DE) algorithm. DE is a simple, but efficient algorithm, which can be applied to solve different types of optimization problems in real-world applications [40]. DE algorithm uses different mutation strategies to exploit the better solutions, but the following mutation strategies are widely adopted.

$$\text{DE/best/1: } V_i = X_{best} + F\,(X_{r1} - X_{r2}) \tag{8}$$

$$\text{DE/rand/1: } V_i = X_{r1} + F\,(X_{r2} - X_{r3}) \tag{9}$$

Where, i = {1, 2, . . . , SN}; r_1, r_2, and r_3 are mutually different random integer indices selected from {1, 2, . . . ,NP};F is a positive real number, typically less than 1.0 that controls the rate at which the population evolves. It is observed that best solution component in the search mechanism can explore the optimum solution effectively and also guide searching process efficiently and resulted in improved convergence rate. In DE algorithm, "DE/best/1", denotes the best solutions explored in the history and it can use to guide the current population, whereas, "DE/rand/1" maintains population diversity. Hence, in this work, a new improved search mechanism is proposed motivated by DE algorithm and new solution search equation is proposed by hybridizing the two solution search equations. The proposed equation is described as follows.

$$X_{k,new} = rand_1 * Z_a * \frac{F_{total}}{m_k} * \Delta t^2 + rand_2 * Z_v * V_{kold} * \Delta t + X_{best} + \varphi * (X_{kold} - X_{r1}) \tag{10}$$

In above equation, the best solution in current population can improve the convergence rate of MCSS algorithm and X_{r1} denotes the random population, maintains the population diversity. Thus, the exploitation ability is improved by using global best experience of each CP in solution search space which is mentioned in equation 10 and this modification leads the algorithm to be converged on optimal solution.

3.2 Local Search Strategy

Every meta-heuristic algorithm suffers with the problem of local optima. Another issue related to these algorithms is the boundary constraints. It is noted that there is no predefined mechanism to deal with such problems. Hence, in this work, effort is made to handle the problems of local optima and boundary constraints. So, a local search strategy is proposed to handle these issues. This strategy is illustrated in Figure 6.1 and summarized as follows:

Step 1: For each CP, Euclidean distance is calculated from the other CPs in the input search space using the position of CP. The other CPs within a threshold Euclidean distance can be considered as neighbors of current CP. Euclidean distance between two CP X_i and X_j in the n-dimensional search space is given by equation 11.

$$d_{ij} = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \quad (11)$$

Step 2: If CP new position is out of range, other CPs in the neighborhood are evaluated.
Step 3: The position of the CP is then replaced with that of the best CP in the neighborhood instead of a random value. This helps in exploring more promising candidate solutions.

So, in this work, two amendments in MCSS algorithm are proposed. First, the solution search equation of the MCSS algorithm is modified using the cognitive equation of PSO algorithm. Second, a neighborhood search strategy is induced in MCSS algorithm to explore the more promising positions of CPs in solution space. The proposed algorithm maintains the core concept of the MCSS algorithm. The detailed description of proposed algorithm is summarized in section 3.1.

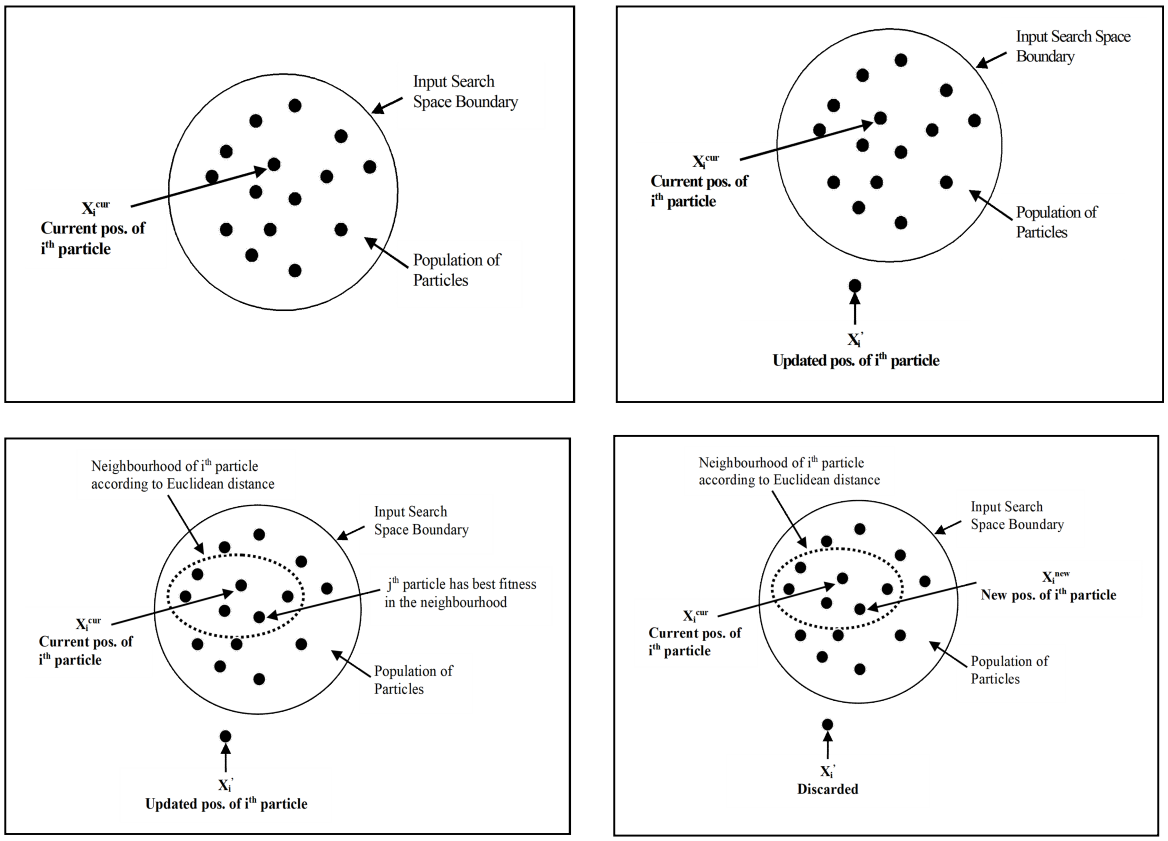

Fig. 1: Local search strategy

3.3 Pseudo code of proposed algorithm

The pseudo code of proposed algorithm is as follows.

Step 1: Initialize the parameters of MCCS algorithm such as number of CPs, c_1, c_2, ϵ and load the dataset.
Step 2: Initialize the charge particles positions and velocities.
For k = 1 to K* K is number of cluster centers in a dataset */
Evaluate the position of k^{th} CP (position (k)) from dataset using equation 12;

$$X(k) = X_{min,i} + r_i * \left((X_{i,max} - X_{i,min}\right) \quad (12)$$

Initialize the velocity of each k^{th} CP;

V (k) = 0;

End for

Step 3: Calculate the value of objective function.

For k = 1: K

Compute the objective function value for each data instance to each CPs using eq. 13;

$$D_{k,i} = \sum_{k=1}^{K} \sum_{j=1}^{N} \sum_{i=1}^{n} \sqrt{\left\| X_{j,i} - X_{k,i} \right\|^2} \quad (13)$$

where, $X_{j,i}$ denotes the i^{th} attribute of the j^{th} data instance, $X_{k,i}$ represents the i^{th} attribute of the k^{th} CPs of instance and $d_{j,k}$denotes Euclidean distance between j^{th} data instance from the k^{th} CPs.

End for

Assign the data instance to each CP using minimum objective function values.

For k = 1: K

Compute the fitness for each CPs using equation 14;

$$D_{k,i} = \sum_{k=1}^{K} \sum_{j=1}^{N} \sum_{i=1}^{n} X_{j,i} - X_{k,i} \quad (14)$$

where, $X_{j,i}$ denotes the i^{th} attribute of the j^{th} data instance, $X_{k,i}$ represents the i^{th} attribute of the k^{th} CPs of instance and $D_{k,j}$denotes distance between k^{th} CPs and j^{th} data instance.

End for

Step 4: For each charge particle

- Initialize the personal best memory

For k=1: K

pbest (k) = position (k);

fitness (pbest) = fitness (k);

End for

- Determine the positions of neighborhood particles in solution space.

For k=1 to K

Compute distance between pbest (k) to every particle.

Evaluate the positions of neighborhood particles (P) to a CP using minimum distance criteria.

End for

Step 5: While the termination condition is not met, do following;

Iteration = 0;

Step 6: Calculate the mass of initial positioned CP.

For k = 1: K

Compute the mass of each CPs using equation 15;

$$M_k = \frac{\text{fit(k)} - \text{fit(worst)}}{\text{fit(best)} - \text{fit(worst)}} \quad (15)$$

Where, m_k represents the mass of k^{th} CPs, fit (k) is the fitness of k^{th} CPs, fit (best) and fit (worst) are the best and worst fitness values in the given dataset.

End for

Step 7: Evaluate the value of moving probability $P_{k,i}$ for each charged particle.

For k=1 to K

Determine the of fitness of each data instance (q_{ik}) to each CP using equation 16;

$$q_{ik} = \frac{\text{fit(i)} - \text{fit(best)}}{\text{fit(k)} - \text{fit(i)}} \quad (16)$$

where, fit(i) presents the fitness of the ith particle while fit(best) and fit(worst) denote the best and worst fitness values of the given dataset.

End for

- Determine the moving probability for each CP.

For k=1 to K

If (fit ($q_{k,i}$) > fit (k))

$P_{k,i}$→ 1;

Else

$P_{k,i}$→ 0;

End if

End for

Step 8: Compute the Electric Force (E_k) for each CP.

- Compute the fitness of data instances.

For i = 1: N * N is total number of data instances in the dataset*/

Compute the fitness of each instance (q_i) using equation 17;

$$q_i = \frac{fit(i) - fit(worst)}{fit(k) - fit(i)}, i = 1, 2, 3, \dots, N \quad (17)$$

End for

- Compute the separation distance (r_{ki}) for each CP.

For k=1 to K

Calculate the value of separation distance (r_{ki}) using equation 18;

$$r_{ki} = \frac{\|X_i - X_k\|}{\|(X_i + X_k)/2 - X_{best}\| + \epsilon} \quad (18)$$

Where, X_i and X_k represent the position of $i^{th}$and k^{th} particle and X_{best}describes the best position and ϵ is a small positive constant to avoid singularities.

End for

If (r_{ki}<R)

w_1→1;

Else

w_2→0;

End if

If (r_{ki}≥ R)

w_1→0;

Else

w_2→1;

End if

- Determine the Electric Force for each CP.

For k=1 to K

Calculate the value of electric force using equation 3;

End for

Step 9: Compute the Magnetic Force (M_{ik}).

- Compute the average electric current (I_i).

For i = 1: N (N is total number of data instances in the dataset)

Determine the average electric current for each data instance (q_i) using equation 19;

$$I_{in} = fit_n(i) - fit_{n-1}(i) \quad (19)$$

End for

- Compute the probability of magnetic influence (PM_k).

For k=1:K

If fit (k) >= fit (i); /* fit (i) represents the fitness of each data instance*/
PM→1;
Else
PM→0;
End if
End for

- Determine the Magnetic Force for each CP.

For k=1 to K
Calculate the value of electric force using equation 4;
End for

Step10: Compute the total force (F_{total}) act on each CPs.

If (rand ( ) > 0.10*(1-iteration / iteration $_{max}$))
Pr=1;
Else
Pr= -1;
End if
For i=1:K
F_{total}= Pr*E_{ik}+M_{ik};
End for

Step 11: Evaluate the new positions$(X_{k,new})$ and velocities $(V_{knew})$of CPs.

For k=1 to K
Update the position of CP using equation 10;
Update the velocity of CP using equation 7;
End for

Step 12: Recomputed the objective and fitness function using updated positions and velocities of CPs.

Step 13: Update the personal best (pbest) position of CPs.

If (fitness (pbest) ≤ fitness ($X_{k,new}$))
fitness (pbest) → fitness ($X_{k,new}$);
Else
pbest (k) = position (k);
End if
For k=1 to K
pbest (k) = position ($X_{k,new}$);
End for

Step 14: * Local Search Strategy*/

- Determine the positions of neighborhood particles in solution space.

For k=1 to K
Compute distance between pbest (k) to every particle.
Evaluate the positions of neighborhood particles (P) to a CP using minimum distance criteria.
End for

- Repositioning of a neighborhood particle as a charge particle (CP)

For k=1 to K
If fitness (k) < rand ()
Select a particle from the neighborhood in random order;
Mark the particle as the new charge particle (CP);
Compute the velocity of new CP using equation 7;
End if
End for

Iteration = Iteration +1;

Step 15: Repeat the step 6 -14, until termination condition is not met.
End while

Step 16: Optimal cluster centers obtained.

3.4 Illustrative example

This section elaborates the process of the proposed method in detail with example dataset. Here, a two dimensional dataset consisting 10 instances is considered for obtaining the optimal cluster centers as well as to explain the operation of the proposed method. The instances of the example dataset are (5.4, 3.7), (4.8, 3.4), (4.8, 3.0), (4.3, 3.0), (5.8, 4.0), (5.7, 4.4), (5.4, 3.9), (5.1, 3.5), (5.7, 3.8) and (5.1, 3.8). These instances are assumed to be divided into three clusters, i.e., K = 3, d = 2, where K is the number of clusters and d is the dimension of data set. The aim is to reduce the intra-cluster distance of the data set. The major steps of the proposed algorithm are as follows.

Step 1 and 2: The proposed algorithm starts by initializing the user defined parameters and identifying the initial cluster centers positions and velocities. Equation 12is used to determine the positions of initial cluster centers (CPs) in random order and it is assumed that the initial velocities of CPs are set to 0. Thus, the randomly generated initial cluster centers are (4.9653, 3.7548), (5.0328, 3.4162) and (5.4318, 3.9185).

Step 3: In step 3, all data instances are grouped into three clusters using the initial cluster centers (4.9653, 3.7548), (5.0328, 3.4162) and (5.4318, 3.9185) by using equation 13. As a result of this, the data instances 1, 2, 3, 4 belong to the cluster center (4.9653, 3.7548), data instances 5, 6, 7 and 8 belong to the cluster center (5.0328, 3.4162) and data instances 9 and 10 belong to the cluster center (5.4318, 3.9185). After grouping the data instances, the fitness of each cluster center is computed using equation 14. The fitness of CPs is 0.1538, 0.0732 and 0.1086.

Step 4: In this step, global best position of CP and its fitness is computed. The global best position of CP and fitness is (5.0328, 3.4162) and 0.0732.The neighbors of current CPs are also evaluated, which are further used to explore the more promising solutions. Thus the neighborhood particles to the current CPs are (4.8, 3.0),(5.1,3.5) and (5.7, 3.8).

Step5, 6 and 7: Step 5 deals with the termination condition. In step 6, mass of initially positioned CPs are calculated using equation 15, which are further used to determine the electric and magnetic force enforced on CPs. Thus, the mass of CPs are 0.4183, 0.7246 and 1.5148. Step 7 is used to determine the moving probability for each CPs in binary form.

Step 8 and 9: In step 8 and 9, the electric and magnetic force acting on each CP is determined using equations3 and 4. These forces can be viewed as the local search for the solutions in solution space and further used with search equation to obtain the optimal cluster centers. Thus, the values of the electric force for CPs are 0.2163, 0.3854 and 0.3189 whereas the magnetic force acting on CPs are 0.0472, 0.0236and 0.0576 respectively.

Step 10: The total force acted on a CP is the combined effect of the electric and magnetic force. But a probabilistic aspect is added to the electric force to give more practical resemblance with actual scenario whether an electric force is attractive or repulsive. If, it is attractive then it will be added to the magnetic force otherwise it will be subtracted. Thus, the values of total force for CPs in example dataset are 0.2348, 0.3972 and 0.2917.

Step 11and 12: The new positions and velocities of CPs are determined using the equations 10 and 7. The updated positions and velocities of CPs are (4.8312 3.6425, 5.1682 3.76963, 5.4318 4.1364) and (0.1543, 0.2381 and 0.1029). In step 12, objective and fitness function values are computed again using the updated positions of CPs.

Step 13: To update the best position of CPs, a comparison between fitness of best position (gbest) and fitness of newly generated positions is performed, the better one take over the gbest.

Step 14 and 15: In this step, a neighborhood search strategy is invoked by identifying the particles near to the current CPs using minimum distance criteria. The neighborhood particles replace the current CPs using degree of randomness and act as new CPs instead of old CPs. Step 14 and 15 are repeated as long as stopping condition is not met and obtain the best cluster centers as an output. Thus, the optimal cluster centers obtain for example dataset using proposed method are (4.5978, 3.1632), (5.2543, 3.8013) and (5.7468, 4.1367).

4.0 EXPERIMENTAL RESULTS

To test the effectiveness of the proposed algorithm, it is applied to solve the clustering problems and results are compared with K-Means, GA, PSO, ACO, CSS, MCSS, and MCSS-PSO algorithms. The proposed algorithm is implemented in Matlab 2010a environment using windows 7 operating system, Intel corei3 processor, 3.4 GHz and 4 GB RAM.

4.1 Performance metrics

The performance of the proposed algorithm is evaluated using the sum of intra cluster distance, and f-measure parameters. These parameters are described as below.

- Intra cluster distances

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. the minimum is the intra cluster distance, the better will be the quality of the solution. The results are measured in terms of best, average, and worst solutions.

- F-Measure

The value of this parameter is measured in terms of recall and precision of an information retrieval system. It is also described as weighted harmonic mean of recall and precision. It can be calculated as if a cluster j consists a set of n_j data objects as a result of a query and each class i consists of a set of n_i data objects need 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 is defined as:

$$\text{Recall}\big(r(i,j)\big) = \frac{n_{i,j}}{n_i} \text{and} \text{Precision}\big(p(i,j)\big) = \frac{n_{i,j}}{n_j} \tag{18}$$

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

$$F(i,j) = \frac{2 * (\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})} \tag{19}$$

The value of F-measure for a given clustering algorithm which consist of n number of data instances is given as

$$F(i,j) = \sum_{i=1}^{n} \frac{n_i}{n} * \max_i * F(i,j) \tag{20}$$

4.2 Parameters settings

The proposed algorithm consists of five parameters $rand_1$, $rand_2$, ϵ, φ, no. of CPs and p in which $rand_1$ and $rand_2$ are random number in the range of 0 and 1, number of CPs is equal to numbers of clusters (K) present in the dataset, ϵ is a small positive constant to avoid singularities and p denotes the number of neighbors. Table 1 shows the parameters settings of the proposed algorithm. In order to achieve a good performance, experiment run with the best parameters values and outcome of experiment measures after 100 iterations. The results presented in this work are the average of ten different runs.

Table 1: Parameter settings of proposed algorithm

Parameters	Values
No. of CPs	No. of Clusters (K)
ϵ	0.001
p	3

4.3 Datasets

Eight real datasets are used to assess the performance of the proposed algorithm. These datasets are taken from the UCI repository. These are iris, wine, CMC, glass, cancer, liver disease (LD), thyroid, and vowel. In spite of these, two artificial datasets are also used to test effectiveness of the proposed algorithm, named as ART1 and ART2 datasets. Table 2 summarizes the features of these datasets.

Table 2: Description of datasets

Dataset	Clusters (K)	Features	Total Instances	Instance in Each Cluster
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)

Tables 3-4 show the results of proposed algorithm and other algorithm being compared. From results, it is stated that the proposed algorithm performs than other algorithms. The proposed algorithm obtains better avg. intra cluster distances for all dataset except thyroid dataset. Further, it is also seen that proposed algorithm obtains best intra cluster distances for all datasets. FM parameter also supports the capability of the proposed algorithm for solving the clustering problems. So, it can be concluded that the proposed clustering algorithm is more competitive and efficient as compared to the other algorithms included in the experiment. To show the effectiveness of the proposed algorithm, some statistical tests are also performed.

Table 3: Comparison of proposed HMCSS algorithm to other meta-heuristics algorithms

Dataset	Para meters	K-means	GA	PSO	ACO	HMCSS
ART1	Best	157.12	154.46	154.06	154.37	155.91
	Avg.	161.12	158.87	158.24	158.52	156.83
	Worst	166.08	164.08	161.83	162.52	158.46
	FM	99.14	99.78	100	100	100
ART2	Best	743	741.71	740.29	739.81	736.63
	Avg.	749.83	747.67	745.78	746.01	741.68
	Worst	754.28	753.93	749.52	749.97	745.84
	FM	98.94	99.17	99.26	99.19	99.57
Iris	Best	97.33	113.98	96.89	97.1	96.36
	Avg.	106.05	125.19	97.23	97.17	96.45
	Worst	120.45	139.77	97.89	97.8	96.93
	FM	0.78	0.78	0.78	0.78	0.79
Wine	Best	16555.68	16530.53	16345.96	16530.53	16028.54
	Avg.	18061	16530.53	16417.47	16530.53	16378.42
	Worst	18563.12	16530.53	16562.31	16530.53	16618.36
	FM	0.52	0.52	0.52	0.52	0.53
LD	Best	11397.83	532.48	209.15	224.76	204.81

	Avg.	11673.12	543.69	224.47	235.16	221.26
	Worst	12043.12	563.26	239.11	256.44	230.73
	FM	0.47	0.48	0.49	0.49	0.49
Cancer	Best	2999.19	2999.32	2973.5	2970.49	2904.65
	Avg.	3251.21	3249.46	3050.04	3046.06	2928.48
	Worst	3521.59	3427.43	3318.88	3242.01	2963.41
	FM	0.83	0.82	0.82	0.82	0.87
CMC	Best	5842.2	5705.63	5700.98	5701.92	5642.61
	Avg.	5893.6	5756.59	5820.96	5819.13	5645.27
	Worst	5934.43	5812.64	5923.24	5912.43	5712.58
	FM	0.33	0.32	0.33	0.33	0.37
Thyroid	Best	13956.83	10176.29	10108.56	10085.82	9956.48
	Avg.	14133.14	10218.82	10149.7	10108.13	10016.78
	Worst	146424.21	10254.39	10172.86	10134.82	10096.74
	FM	0.73	0.76	0.78	0.78	0.8
Glass	Best	215.74	278.37	270.57	269.72	210.08
	Avg.	235.5	282.32	275.71	273.46	215.45
	Worst	255.38	286.77	283.52	280.08	224.34
	FM	0.43	0.33	0.36	0.36	0.46
Vowel	Best	149422.26	149513.73	148976.01	149395.6	142326.74
	Avg.	159242.89	159153.49	151999.82	159458.14	144012.39
	Worst	161236.81	165991.65	158121.18	165939.82	154286.97
	FM	0.65	0.65	0.65	0.65	0.66

Table 4: Performance comparison of proposed HMCSS to other CSS variants

Dataset	**Para meters**	**CSS**	**MCSS**	**MCSS-PSO**	**HMCSS**
ART1	Best	153.91	153.18	152.76	155.91
	Avg.	158.29	158.02	157.24	156.83
	Worst	161.32	159.26	158.92	158.46
	FM	100	100	100	100
ART2	Best	738.96	737.85	738.19	736.63
	Avg.	745.61	745.12	742.34	741.68
	Worst	749.66	748.67	746.58	745.84
	FM	99.43	99.56	99.56	99.57
Iris	Best	96.47	96.43	96.31	96.36
	Avg.	96.63	96.57	96.48	96.45
	Worst	96.78	96.63	96.65	96.93
	FM	0.79	0.79	0.79	0.79
Wine	Best	16282.12	16158.56	16047.32	16028.54
	Avg.	16289.42	16189.96	16093.18	16078.42
	Worst	16317.67	16223.61	16128.33	16108.36
	FM	0.53	0.54	0.54	0.53
LD	Best	207.09	206.14	203.71	204.81
	Avg.	228.27	221.69	219.69	221.26

	Worst	242.14	236.23	231.47	230.73
	FM	0.49	0.5	0.5	0.49
Cancer	Best	2946.48	2932.43	2918.93	2904.65
	Avg.	2961.16	2947.74	2931.56	2928.48
	Worst	3006.14	2961.03	2978.29	2963.41
	FM	0.85	0.86	0.86	0.87
CMC	Best	5672.46	5653.26	5638.64	5642.61
	Avg.	5687.82	5678.83	5654.37	5645.27
	Worst	5723.63	2947.74	5691.43	5712.58
	FM	0.36	0.37	0.37	0.37
Thyroid	Best	9997.25	9963.89	9917.56	9956.48
	Avg.	10078.23	10036.93	10023.08	10016.78
	Worst	10116.52	10078.34	10104.26	10096.74
	FM	0.79	0.79	0.79	0.8
Glass	Best	203.58	209.47	206.32	210.08
	Average	223.44	231.61	217.61	215.45
	Worst	241.27	263.44	226.44	224.34
	FM	0.45	0.45	0.45	0.46
Vowel	Best	149335.61	146124.87	142948.63	142326.74
	Avg	152128.19	149832.13	145640.78	144012.39
	Worst	154537.08	157726.43	153568.24	154286.97
	FM	0.65	0.65	0.65	0.66

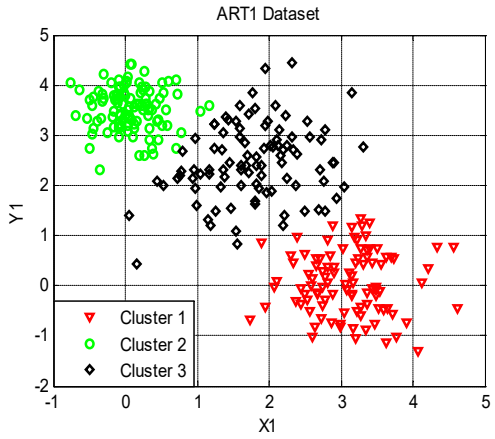

Fig. 2 : Clustering of ART1 dataset using proposed algorithm

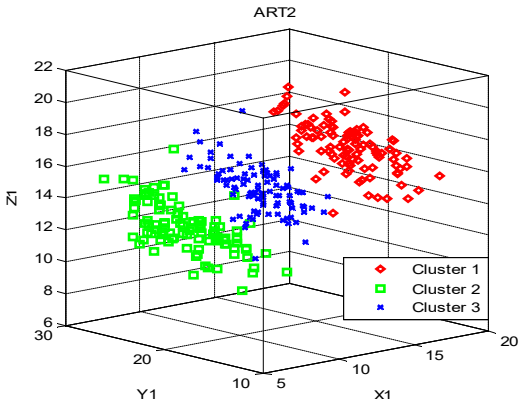

Fig. 3 : Clustering of ART2 dataset using proposed algorithm

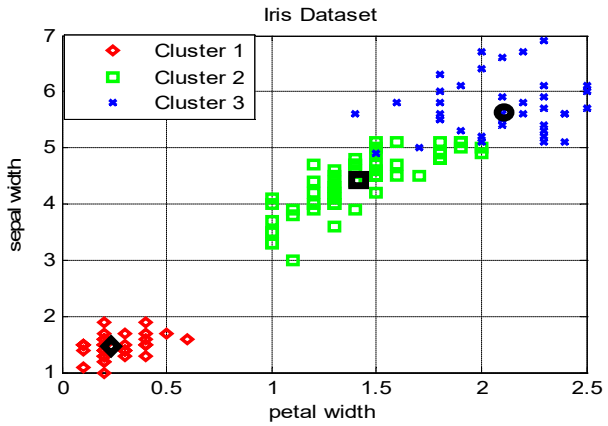

Fig. 4 : Clustering of iris dataset using proposed approach in 2Dimension

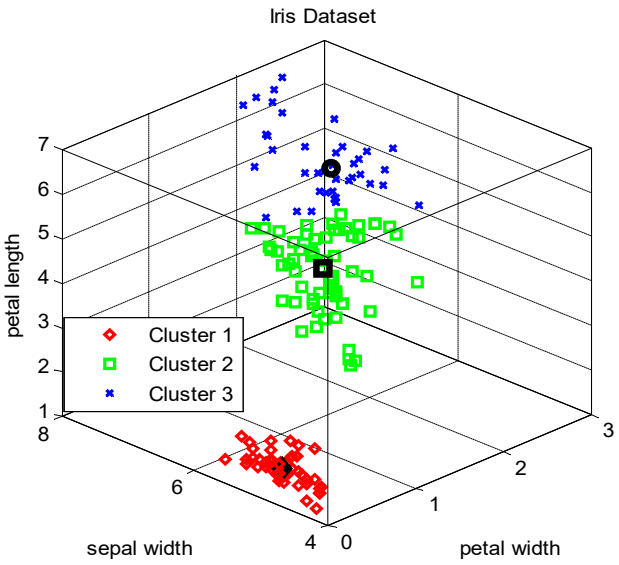

Fig. 5 : Clustering of iris dataset using proposed approach in 3Dimension

ART1 and ART2 data are arranged into three different clusters using the proposed algorithm and the arrangement of data into different clusters is shown in fig. 2 and 3. The ART1 dataset consists of two dimensions , wheras, ART2 dataset consists of three dimensions. Further, fig. 4 and 5 shows the arrangmnet of iris data into three different clusters. Fig. 4 illustrates the data using sepal width and petal width attributs, wheras, fig 5 shows the arrangement of data using three dimension such as petal length, sepal width and petal width. From these figs., it is also concluded that data belongs to cluster 1 is linearly separable from other two clusters. But the data belongs to clusters 2 and 3 are linearly non separable.

4.4 Statistical analysis

This subsection section deals with the statistical analysis on the performance of the proposed HMCSS algorithm. The objective of statistical analysis is to find the significant differences between the performances of all algorithms. These tests have been widely applied in the machine learning domain [41-43]. Statistical analysis consists of two steps. In first step, a statistical test is applied to check the substantial differences in the outcomes of the algorithms. In second step, a post-hoc test is applied to validate the best performing algorithm. In post-hoc test, best performing algorithm acts as a control algorithm and its performance is measured with the rest of the algorithms. In this work, Friedman and Quade tests are employed for demonstrating the statistical analysis and along these tests, holm's method is selected as post-hoc test and level of confidence (α) is set as 0.05.

Table 5: Average ranking of algorithm using Friedman test using intra cluster distance parameter

Algorithms	**K-Means**	**GA**	**PSO**	**ACO**	**CSS**	**MCSS**	**MCSS-PSO**	**HMCSS**
Ranking	7.5	6.85	5.4	5.95	4.2	3.1	1.9	1.1

Results of the Friedman test are illustrated in Table 5 and 8. Table 5 summarizes the average ranking of each technique computed using Friedman test for intra cluster distance parameter. The statistical value of Friedman test on the confidence level 0.0.5 is 63.0834 and corresponding p-value is 3.647e-11. The null hypothesis is clearly rejected indicating a significant difference in the performance of all algorithms. But, it is noticed that Friedman test cannot differentiate the datasets that are used in experimentation and it assigns equal importance to each of the dataset. However, varieties of datasets are used in this work and these are categorized into three categories, which are low dimensional datasets, moderate dimensional datasets, and high dimensional dataset. For instance, iris dataset is one of the low dimensional dataset, whereas wine dataset corresponds to the high dimensional dataset. So, an alternative approach is used for ranking of algorithms on the basis of scaling of datasets and it provides more accurate results in comparison to Friedman test.

An alternative approach to Friedman test is Quade Test. Tables 6 to 8 illustrates the results of Quade test using intra cluster distance parameter. The relative size of observations with each algorithm is shown in Table 6 and it is used to determine the relative significance of the datasets. Table 7 shows the average ranking of algorithms using Quade test. The results of Quade test are shown in Table 8 and the statistical values measured on the confidence level 0.0.5 are 29.7752and corresponding p-value is 1.306e-10. Again the null hypothesis is rejected at the level of confidence 0.05. Both methods rejected the null hypothesis and it can be concluded that substantial differences exist among the performance of all algorithms being compared.

Table 6: Relative size of each dataset for Quade test using intra cluster distance parameter

Dataset	Algorithm							
	K-Means	**GA**	**PSO**	**ACO**	**CSS**	**MCSS**	**MCSS- PSO**	**HMCSS**
Iris	12.5	17.5	7.5	2.5	-2.5	-7.5	-12.5	-17.5
Wine	3.5	2	0.5	2	-0.5	-1.5	-2.5	-3.5
LD	17.5	12.5	-2.5	7.5	2.5	-7.5	-17.5	-12.5
Cancer	17.5	12.5	7.5	2.5	-2.5	-7.5	-12.5	-17.5
CMC	17.5	2.5	12.5	7.5	-2.5	-7.5	-12.5	-17.5
Thyroid	17.5	12.5	7.5	2.5	-2.5	-7.5	-12.5	-17.5
Glass	2.5	17.5	12.5	7.5	-7.5	-2.5	-12.5	-17.5
Vowel	12.5	7.5	-2.5	17.5	2.5	-7.5	-12.5	-17.5
Relative Size of datasets	101	84.5	43	49.5	-13	-49	-95	-121

Holm's post-hoc test is performed after the rejection of null hypothesis. This test can be done for illustrating the significant differences occur between best one and rest of algorithms. The best algorithm can act as the benchmark algorithm and its performance is measured against the other algorithms. Here, HMCSS algorithm acts as a control algorithm and the results of the post-hoc test with both of Friedman and Quade tests are summarized in Table 9 to10. From these, it can be observed that post-hoc test rejects the hypothesis at the confidence level of 0.1 and 0.05 except MCSS-PSO. So, it can be proven that there is a significant difference in the performance of algorithms. The null hypothesis is not rejected with MCSS-PSO algorithm but, it is noted that the performances of both algorithms are significantly different.

Table 7: Ranking of each algorithm computed through Quade test using intra cluster distance parameter

Algorithm	**K-Means**	**GA**	**PSO**	**ACO**	**CSS**	**MCSS**	**MCSS-PSO**	**HMCSS**
Avg. Ranking	7.38	6.81	5.63	5.94	4.13	3.13	1.88	1.13

Both of tests are also applied on FM parameter results. Results of these tests are shown in Table 11to14. Table 11 shows the average ranking of each algorithm computed using Friedman tests for f-measure parameter. It is noticed that HMCSS algorithm obtains 1st rank while GA gets the worst rank among all algorithms.

Table 8: Statistics obtained using Friedman and Quade tests for intra cluster distance parameter.

Method	Statistical Value	p-value	Hypothesis
Friedman	63.0834	3.647e-11	Rejected
Quade	29.7752	1.306e-10	Rejected

The statistical value obtained for Friedman test at the confidence level 0.05 is 45.3741and the statistics of Friedman test is summarized in Table 14 which rejects the null hypothesis. Tables 12 to14 consist of the outcomes of the Quade test. Table 12 summarizes the relative size of each dataset for each algorithm, whereas the ranking of each algorithm using Quade test is illustrated in Table 13. Statistics of Quade test is shown in Table 14 and the statistical value on the confidence level 0.05 is 14.2635. It can be said that null hypothesis is rejected at the level 0.05.

Table 9: Results of Holm's Post-hoc test for Friedman Test using intra cluster distance parameter

i	Algorithms	z values	p values	α/i, α=0.05	Hypothesis	α/i, α=0.1	Hypothesis
7	K-Means	5.3127	< 0.00001	0.00714	Rejected	0.01429	Rejected
6	GA	4.8643	1.4636E-06	0.08330	Rejected	0.01667	Rejected
4	PSO	3.9286	5.8954E-05	0.01000	Rejected	0.02000	Rejected
5	ACO	3.6352	4.6324E-05	0.01250	Rejected	0.02500	Rejected
3	CSS	2.3296	0.02180	0.01666	Rejected	0.03333	Rejected
2	MCSS	1.8329	0.01928	0.25000	Rejected	0.05000	Rejected
1	MCSS-PSO	0.9134	0.17612	0.05000	Not Rejected	0.10000	Not Rejected

Table 10: Results of Holm's Post-hoc test for Quade Test using intra cluster distance parameter

i	Algorithms	z values	p values	α/i, α=0.05	Hypothesis	α/i, α=0.1	Hypothesis
7	K-Means	5.5783	< 0.00001	0.00714	Rejected	0.01429	Rejected
6	GA	5.1262	4.6315E-07	0.08330	Rejected	0.01667	Rejected
4	PSO	4.5873	3.6648E-06	0.01250	Rejected	0.02500	Rejected
5	ACO	3.7219	1.7326E-04	0.01000	Rejected	0.02000	Rejected
3	CSS	2.5493	0.00836	0.01666	Rejected	0.03333	Rejected
2	MCSS	2.1357	0.02618	0.25000	Rejected	0.05000	Rejected
1	MCSS-PSO	0.9126	0.1735	0.05000	Not Rejected	0.10000	Not Rejected

Table 11: Average ranking of algorithm using Friedman tests for FM parameter

Algorithms	K-Means	GA	PSO	ACO	CSS	MCSS	MCSS-PSO	HMCSS
Ranking	6.25	6.88	6	6	3.75	2.75	2.75	1.63

Table 12: Relative size of each dataset computed through Quade test for FM parameter

Dataset	Algorithm							
	K-Means	GA	PSO	ACO	CSS	MCSS	MCSS-PSO	HMCSS
Iris	3	3	3	3	-3	-3	-3	-3
Wine	6	6	6	6	-3	-9	-9	-3
LD	24.5	17.5	3.5	3.5	3.5	-14	-14	-24.5
Cancer	2.25	11.25	11.25	11.25	-2.25	-9	-9	-15.75
CMC	6.75	15.75	6.75	6.75	-2.25	-11.25	-11.25	-11.25
Thyroid	24.5	17.5	7	7	-10.5	-10.5	-10.5	-24.5
Glass	3.5	24.5	14	14	-10.5	-10.5	-10.5	-24.5
Vowel	0.75	0.75	0.75	0.75	0.75	0.75	0.75	-5.25
Relative Size of datasets	71.25	96.25	52.25	52.25	-27.25	-66.5	-66.5	-111.75

Table 13: Ranking of each algorithm determined through Quade test using FM parameter

Algorithm	K-Means	GA	PSO	ACO	CSS	MCSS	MCSS-PSO	HMCSS
Avg. Ranking	6.25	6.88	6	6	3.75	2.75	2.75	1.63

Table 14: Statistics of Friedman and Quade tests using FM parameter

Method	Statistical Value	p-value	Hypothesis
Friedman	45.3741	1.157e-7	Rejected
Quade	14.2635	7.489e-10	Rejected

Again, Holm's post hoc test is to prove substantial difference between best one i.e. control algorithm and other algorithms. Results of post-hoc method are described in Tables 15 to16 using intra cluster distance and FM parameters at the level of confidence 0.05 and 0.1. From Table 16, it can be revealed that hypothesis is rejected with most of algorithms except MCSS-PSO at the confidence level 0.05. While at the confidence level 0.1, hypothesis is rejected with all of algorithms except MCSS-PSO which signifies that performance of the control algorithm MCSS-PSO considerable is diversified from the others. In case of MCSS-PSO, when $\alpha = 0.05$ and MCSS-PSO when $\alpha = 0.1$, it is mentioned that the performance of the proposed method is better than the other algorithms as reported in Table 4. Whereas, the results of the Holm's post-hoc test for Quade test is shown in Table 16. From this it can be concluded that the proposed algorithm outperforms others at the significance level 0.1 and rejects all hypothesis. At the significance level 0.05, the proposed algorithm rejects the hypothesis except MCSS-PSO. It can be concluded that performance of the proposed

algorithm is significantly different from the others and statistical analysis study also proves the substantial significance of the proposed algorithm.

Table 15: Results of Holm's Post-hoc test after rejection of null hypothesis for Friedman Test using FM.

i	Algorithms	z values	p values	α/i, α=0.05	Hypothesis	α/i, α=0.1	Hypothesis
7	K-Means	3.5263	1.3265E-04	0.00714	Rejected	0.01429	Rejected
6	GA	5.6193	< 0.00001	0.08330	Rejected	0.01667	Rejected
5	ACO	4.3365	1.4158E-05	0.01000	Rejected	0.02000	Rejected
4	PSO	4.1832	2.3283E-05	0.01250	Rejected	0.02500	Rejected
3	CSS	2.7563	0.00836	0.01666	Rejected	0.03333	Rejected
2	MCSS	1.5749	0.09412	0.25000	Rejected	0.05000	Rejected
1	MCSS-PSO	1.0503	0.17367	0.05000	Not Rejected	0.10000	Not Rejected

Table 16: Results of Holm's Post-hoc test after rejection of null hypothesis for Quade Test using FM.

i	Algorithms	z values	p values	α/i, α=0.05	Hypothesis	α/i, α=0.1	Hypothesis
7	GA	5.8371	< 0.00001	0.0833	Rejected	0.01429	Rejected
6	ACO	4.9637	1.8632E-06	0.0100	Rejected	0.01667	Rejected
5	PSO	4.6374	6.8584E-06	0.0125	Rejected	0.02000	Rejected
4	K-Means	4.5453	13196E-05	0.0071	Rejected	0.02500	Rejected
3	CSS	2.8425	0.00574	0.0167	Rejected	0.03333	Rejected
2	MCSS	1.6256	0.09256	0.2500	Rejected	0.05000	Rejected
1	MCSS-PSO	1.4238	0.07635	0.0500	Not Rejected	0.10000	Rejected

5.0 CONCLUSION

In this work, a hybrid magnetic charge system search (HMCSS) is proposed for efficient data clustering. In the proposed algorithm, a modified solution search equation is suggested using DE algorithm to enhance the exploitation ability of the MCSS algorithm. Moreover, a concept of local search strategy is also incorporated into the proposed MCSS-PSO algorithm to explore the more promising solutions and also to handle boundary constraints. The designed concept and detailed procedure are presented with the help of illustrative example. The proposed algorithm is also capable to overcome the local optima problem and explores the more promising solution direction. The capability of proposed algorithm is tested for solving clustering problems. The experimental results demonstrate the better performance of the

proposed algorithm in comparison to other clustering algorithms being compared. In addition to it, some statistical tests are also performed to prove the efficacy of the proposed algorithm. Finally, it can be concluded that the proposed HMCSS is more efficient and robust algorithm for clustering problems.

REFERENCES

[1] M.R. Anderberg, *Cluster Analysis for Application*, New York, Academic Press, 1973.

[2] J.A. Hartigan, *Clustering Algorithms,* New York, Wiley, 1975.

[3] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data.* New Jersey, Prentice-Hall, 1988.

[4] S. Milan, V. Hlavac, and R. Boyle. *Image processing, analysis, and machine vision*, Champion and Hall, 1998, pp 2-6.

[5] P. Teppola, S. P. Mujunen, and P. Minkkinen, "Adaptive Fuzzy C-Means clustering in process monitoring", *Chemometrics and intelligent laboratory systems*, Vol. 45, No. 1, 1999, pp. 23-38.

[6] A. Webb, *Statistical pattern recognition*, New Jersey, John Wiley & Sons, 2002, pp. 361–406.

[7] H. Zhou, and L. Yonghuai, "Accurate integration of multi-view range images using k-means clustering", *Pattern Recognition,* Vol. 41, No. 1, 2008, pp. 152-175.

[8] E. Alpaydin, *Introduction to machine learning*, MIT press, 2004.

[9] WJ. Dunn III, M.J. Greenberg, and S. C. Soledad, "Use of cluster analysis in the development of structure-activity relations for antitumor triazenes", *Journal of medicinal chemistry*, Vol. 19, No. 11, 1976, pp. 1299-1301.

[10] Y. He, W. Pan, and L. Jizhen, "Cluster analysis using multivariate normal mixture models to detect differential gene expression with microarray data", *Computational statistics & data analysis*, Vol. 51, No. 2, 2006, pp. 641-658.

[11] R. Xu and D.C. Wunsch, *Clustering,* Oxford, Wiley, 2009.

[12] S. Das, A. Abraham, A. Konar, *Meta-heuristic Clustering,* Springer, 2009, ISBN 3540921729.

[13] E.R. Hruschka, R.J.G.B. Campello, A.A. Freitas and A.C.P.L.F. De Carvalho, "A survey of evolutionary algorithms for clustering", *IEEE Transactions on Systems, Man and Cybernetics.*, Vol. 39, No. 2, 2009, pp. 133–155.

[14] A. Abraham and S. Das, S. Roy, "Swarm intelligence algorithms for data clustering", *Soft Computing for Knowledge Discovery and Data Mining*, Springer, 2007, pp. 279–313.

[15] S. Basu, I. Davidson, K. Wagstaff, *Constrained Clustering: Advances in Algorithms, Theory and Applications, Data Mining and Knowledge Discovery,* Chapman and Hall/CRC, 2008.

[16] A.K. Jain, "Data clustering: 50 years beyond K-means", *Pattern Recognition Letters,* Vol. 31, No. 8, 2010, pp 651-666.

[17] J. MacQueen, "Some methods for classification and analysis of multivariate observations", in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability,* Vol. 1, 1967, pp 281-297.

[18] C. A. Murthy and N. Chowdhury, "In search of optimal clusters using genetic algorithms", *Pattern Recognition Letters,* Vol. 17, No. 8, 1996, pp 825-832.

[19] R.J. Kuo and L.M. Lin, "Application of a hybrid of genetic algorithm and particle swarm optimization algorithm for order clustering", *Decision Support Systems,* Vol. 49, 2010, pp 451–462.

[20] C. Y. Chen and F. Ye, "Particle swarm optimization algorithm and its application to clustering analysis", In *IEEE international conference on in Networking, Sensing and Control*, Vol. 2, 2004, pp. 789-794.

[21] T. Niknam and B. Amiri, "An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis", *Applied Soft Computing, Vol.* 10, No. 1, 2010, pp 183–197.

[22] P. S. Shelokar, V. K. Jayaraman, and B. D. Kulkarni, "An ant colony approach for clustering", *Analytica Chimica Acta*, Vol. 509, No. 2, 2004, pp 187-195.

[23] A. N. Sinha, N. Das, and G. Sahoo, "Ant colony based hybrid optimization for data clustering", *Kybernetes*, Vol. 36, No. 2, 2007, pp.175 – 191.

[24] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm", *Applied Soft Computing*, Vol. 11, No. 1, pp 652–657.

[25] X. Yan, Y. Zhu, W. Zou and L. Wang, "A new approach for data clustering using hybrid artificial bee colony algorithm", *Neurocomputing,* Vol. 97, 2012, pp 241–250.

[26] S. C. Satapathy and A. Naik, "Data clustering based on teaching-learning-based optimization", In *Swarm, Evolutionary, and Memetic Computing, Cochin, India*, 2011, pp. 148-156.

[27] A. J. Sahoo, and Y. Kumar, "Modified Teacher Learning Based Optimization Method for Data Clustering", In *Advances in Signal Processing and Intelligent Recognition Systems, Kerla, India,* 2014, pp. 429-437.

[28] Y. Kumar and G. Sahoo, "A charged system search approach for data clustering", *Progress in Artificial Intelligence,* Vol. 2, No. 2-3*,* 2014, pp 153-166.

[29] Y. Kumar and G. Sahoo, "A chaotic charged system search approach for data clustering", *Informatica*, Vol. 38, No. 3, 2014, pp 249-261.

[30] S. Z. Selimand and K. Alsultan, "A simulated annealing algorithm for the clustering problem", *Pattern recognition,* Vol. 24, No. 10, 1991, pp 1003-1008.

[31] U. Maulik and A. Mukhopadhyay, "Simulated annealing based automatic fuzzy clustering combined with ANN classification for analyzing microarray data", *Computers & Operations Research*, Vol. 37, No. 8, 2010, pp 1369-1380.

[32] C. S. Sung and H. W. Jin, "Atabu-search-based heuristic for clustering", *Pattern Recognition,* Vol. 33, No. 5, 2000, pp 849-858.

[33] Y. Kumar and G. Sahoo, "A Hybrid Data Clustering Approach Based on improved Cat Swarm Optimization and K- Harmonic Mean Algorithm", *AI communications*, Vol. 28, No. 4, 2015, pp, 751-764.

[34] Y. Kumar and G. Sahoo, "A Hybridize Approach for Data Clustering Based on Cat Swarm Optimization", *International Journal of Information and Communication Technology*, Vol. 9, No. 1, 2016, pp. 117-141.

[35] S. Budi, and M. K. Ningrum, "Cat swarm optimization for clustering", In *International Conference on Soft Computing and Pattern Recognition (SOCPAR'09),* 2009, pp. 54-59.

[36] Y. Kumar and G. Sahoo, "An improved Cat Swarm Optimization Algorithm for Data Clustering", In *International Conference on Computational Intelligence in Data Mining, Odisha, India*, Vol. 1, 2014, pp. 187-197.

[37] Y. Kumar, S. Gupta, and G. Sahoo, "A Clustering Approach Based on Charged Particle", *International Journal of Software Engineering and Its Applications*, Vol. 10, No. 3, 2016, pp. 9-28.

[38] Y. Kumar and G. Sahoo, "Hybridization of magnetic charge system search and particle swarm optimization for efficient data clustering using neighborhood search strategy", *Soft Computing*, Vol. 19, No. 12, 2015, pp. 3621-3645.

[39] A. Kaveh, A., M. A. M. Share, and M. Moslehi, "Magnetic charged system search: a new meta-heuristic algorithm for optimization",*Acta Mechanica*, Vol. 224, No. 1, 2013, pp 85-107.

[40] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces", *Journal of Global Optimization,* Vol. 23, 2010, pp. 689–694.

[41] M. Moohebat, R.G. Raj, D. Thorleuchter and S. Abdul-Kareem. "Linguistic Feature Classifying and Tracing". Malaysian Journal of Computer Science, Vol. 30, No.2, 2017, pp 77-90.

[42] R.G. Raj and S. Abdul-Kareem, "Information Dissemination And Storage For Tele-Text Based Conversational Systems' Learning", Malaysian Journal of Computer Science, Vol. 22 No. 2, 2009. pp. 138-159.

[43] A. Qazi, R. G. Raj, M. Tahir, M. Waheed, S. U. R. Khan, and A. Abraham, "A Preliminary Investigation of User Perception and Behavioral Intention for Different Review Types: Customers and Designers Perspective," The Scientific World Journal, Vol. 2014, Article ID 872929, 8 pages, 2014. doi:10.1155/2014/872929.