An Efficient Hybrid Algorithm using Cuckoo Search and Differential Evolution for Data Clustering

Asgarali Bouyer, Habib Ghafarzadeh and Omid Tarkhaneh
Azarbaijan Shahid Madani University, Tabriz - 51368, Iran; a.bouyer@azaruniv.edu, Habib.ghafarzadeh@gmail.com

Abstract

Data clustering is a well-known data analysis technique needed in many fields such as Engineering, Education, Medicine, and Bioinformatics. The-means algorithm is one of the most common algorithms used for data clustering. However, the results of K-means highly depend on the initial centers and may converge to local optima. In order to overcome these problems, many studies have been done in clustering. This paper proposes hybrid approach for data clustering using cuckoo search and differential evolution algorithms. Cuckoo Search (CS) is a novel swarm-intelligence based algorithm that has been proposed recently. This algorithm has less control parameters and good ability in solving different kinds of problems, however it has some problems like much number of functional evaluation and it sometimes easily get trapped in local minima. The proposed algorithm benefits from Differential Evolution Algorithm (DE) and also benefits from Mantegna levy distribution to promote the CS algorithm in solving the number of functional evaluation and also obtaining better convergence speed and high precision in a short time. For testing the performance of the proposed algorithm this paper employed six real standard benchmark datasets from UCI machine learning repository. The simulation results show that the proposed algorithm has ability in obtaining better results in terms of Convergence Speed, Accuracy, and also reducing number of functional evaluation.

Keywords: Data Clustering, Convergence Speed, Cuckoo Search, Swarm Intelligence

1. Introduction

In recent years many swarm intelligence-based algorithms have been proposed. Most of them are nature inspired algorithms such as Artificial Bee Colony (ABC)\(^1\), Particle Swarm Optimization (PSO)\(^2\) and Ant Colony Optimization (ACO)\(^3\). Cuckoo Search (CS) is also one of the nature inspired algorithm proposed by Yang and Debin\(^4\). Functional evaluation in one evolutionary algorithm indicates to the number of calls of the objective function. In one evolutionary algorithm a good convergence speed and reaching to the good accuracy and also having less number of functional evaluations is important. The CS algorithm has the problems of much number of functional evaluation and also it sometimes easily get trapped in local minima and this point causes having less convergence speed and also less accuracy. To tackle these problems, in this paper a Hybrid approach proposed by combining CS and Differential Evaluation (DE) (HCSDE). The DE algorithm is also an efficient Meta Heuristic algorithm. DE uses the differences between randomly selected vectors (chromosome and individuals)\(^5\). The hybrid approach in this paper helps CS algorithm to reduce the number of functional evaluations and also obtaining better convergence speed and accuracy. Data Clustering (cluster analysis) is an unsupervised method for making batches of objects in which objects that belong to a particular cluster are similar to each other and are dissimilar from different clusters. Data Clustering is a vital technique for Image Segmentation\(^6\), Fraud Detection\(^7\), Biomedical Data\(^8\), wireless mobile sensor networks\(^9\), and other applications that we encounter in some fields of sciences and engineering. Clustering algorithms is organized in two main types: Hierarchical
and Partitional clustering algorithm. A Partitional algorithm makes one partition from a dataset and is used for huge datasets whereas the hierarchical type makes array of nested partitions. K-means is one type of partitional algorithm that is familiar to most of the researchers who are working on problems related to data clustering issues. Unfortunately k-means algorithm has some problems, first it must know the number of the clusters at the beginning of algorithm, second it is depended on initial centroids and this may send algorithm to the danger of local minima solutions. Using Evolutionary algorithms, we rarely encounter the drawbacks of k-means algorithm. In recent years, many researchers have proposed many evolutionary algorithms for clustering problem. Niknam et al. proposed an evolutionary optimization algorithm based on ACO and Simulated Annealing (SA) to solve the clustering problem. Niknam et al proposed a Hybrid Evolutionary algorithm based on PSO and SA to find optimal cluster centers. Niknam and Amiri proposed an efficient hybrid approach based on PSO, ACO and K-means for cluster analysis. Chi-Yang Tsai et al proposed a new evolutionary algorithm based on PSO with selective particle regeneration for data clustering. Shelokar introduced an Evolutionary algorithm based on ACO algorithm for clustering problem. Paterlini et al. used Differential Evolution (DE) and PSO algorithms to solve the clustering problem. Hatamlou et al. proposed a combined approach for clustering based on K-means and gravitational search algorithms. Yan et al. introduced Hybrid ABC algorithm for data clustering. This paper focused on the partitional clustering algorithm and it uses the proposed algorithm (HCSDE) for it. We test the efficiency of the proposed algorithm by employing six standard benchmark dataset from UCI machine learning repository and we claim that the proposed algorithm is efficient in terms of reducing number of functional evaluation and also obtaining better convergence speed and accuracy. So the objective and motivation of this work is removing the mentioned problems of CS algorithm and also proposing an efficient algorithm for data clustering in comparison with the employed algorithms. The proposed algorithm also can be used for the other optimization problems. The rest of the paper is organized as follows: In section 2 we will debate the clustering problem. In section 3 we will discuss standard CS. In section 4 we will describe standard DE. We will introduce HCSDE in detail in section 5. In section 6 we will test the proposed algorithm on six datasets from UCI for the clustering problem, finally in section 7 the conclusions will be drawn.

2. The Clustering Problem

Clustering aims to find a set of patterns, points or objects. It tries to represent n objects, find k groups based on measure of similarity In which objects that are belongs to the same cluster are similar as possible to each other and objects in different clusters are dissimilar as possible. Generally, similarity and dissimilarity are mutual concepts. Usually, similarity used to describe how similar two data points are or how similar two clusters are. If the similarity coefficient be greater, it means that two clusters and two data points are more similar. Dissimilarity measure or distance is the other way around. The greater dissimilarity or distance indicates to that, the two data points or two clusters are more dissimilar. Let \( O = \{o_1, o_2, ..., o_n\} \) be the set of n objects and the k cluster be represented by \( \{c_1, c_2, ..., c_k\} \). Then:

\[
C_i, j \neq i \text{ for } i, j = 1, 2, ..., k, \text{ and } i \neq j, \text{ and } \bigcup_{i=1}^{k} C_i = O
\]

One of the similarity criteria is the total Mean Square Quantization Error (MSE) Equation (1). MSE is a clustering quality measure and indicates to the total within cluster variance, and aims are minimizing this measure to find the most compact partitioning for the k partitions.

\[
\text{perf}(O,C) = \sum_{i=1}^{n} \min_{j=1,2,...,k} \left\| o_i - c_j \right\|, j=1,2,...,k
\]

Where \( \left\| o_i - c_j \right\|^2 \) shows the similarity between the ith object and the center of the jth cluster. The Euclidean distance is probably the most well known metric that we have ever used which is derived from Minkowski metric Equation (2).

\[
d(o_i, c_j) = \left( \sum_{m=1}^{p} (x_{im} - c_{jm})^2 \right)^{1/2}
\]

Where \( d(o_i, c_j) \) is a function that denotes to a dissimilarity measure between object i and j. In this research we also use Euclidean distance as a distance metric. As mentioned above, we will focus on partitional algorithms. One of the most used partitional classes is the center based type. K-means, as a well known clustering...
algorithm, belongs to this class of partitional algorithms\textsuperscript{14}. The steps of k-means algorithm are as follows\textsuperscript{21}:

1. Choose k cluster centers \((c_1, c_2,.., c_k)\) from n objects \((o_1, o_2,.., o_n)\) randomly.
2. Bind objects \(o_i, i = 1, 2,..., n\) to cluster \(c_j, j = 1, 2,..., k\) if \(|o_i - c_j| < |o_i - c_p|, p = 1, 2,..., k\) and \(j \neq p\).
3. Calculate new cluster centers using the following equation:
\[
c_i^* = \frac{1}{n_i} \sum_{o_j \in c_i} X_j, j = 1, 2,..., k.
\]

Where \(n_i\) is the number of elements that belongs to the cluster \(c_i\).
4. Repeat step 2 and 3 until the termination criteria are satisfied.

K-means is a fast algorithm but it is trapped easily in local minima because of the sensitivity to the initial cluster centers.

3. Cuckoo Search Algorithm

In this section we will introduce Cuckoo Search algorithm briefly.

3.1 Cuckoo Breeding Behavior

Cuckoo is one of brood parasites birds. Yang and Deb by studying the life style of this bird proposed an algorithm called cuckoo search in 2009. This bird has a reproduction strategy for her survive. Cuckoo laid her eggs in nest of other birds called host birds. Some host birds can diagnose and detect cuckoo eggs, so in this case they will do two actions, they may throw the cuckoo eggs out or they may abandon their nest and go to another place to live in. One interesting thing about some species of cuckoos is that, they have evolved in such a way that female parasitic cuckoo scan imitate the colors and patterns of the host bird eggs, doing this action can reduce discovery rate and increase their productivity. In addition, egg laying in some species takes time. In general, the cuckoo eggs hatch faster than their host eggs. When the first cuckoo chick hatches, it evicts the host eggs by blindly propelling the eggs out of the nest. This action helps to increase the cuckoo chicks share provided foods by its host bird. Studies also show that cuckoo chick can mimic the call of host chicks and this action also help to improve feeding opportunity\textsuperscript{4}.

3.2 Levy Flight

The earlier studies show that the foraging path of animal is a random walk, because the next move depends on both, the current location/state and the transition probability to the next location. A random walk is a random activity. It includes taking a series of consecutive random steps. A levy flight is a random walk whose step length obeys the levy distribution. Studies also show that the behaviors of human like ju/hoansi hunter express foraging patterns which are a case of typical features of levy flights. Levy flights have many applications, for example many physical phenomena like cooling behavior and noise, reflect the features of levy flights in right conditions\textsuperscript{22}.

3.2.1 Cuckoo Search Algorithm Implementation

Cuckoo search is based on three idealized rules:

- Each cuckoo lays one egg at a time, and dumps it in a nest which was chosen randomly.
- The nests with high fitness of eggs (solutions) are transferred to the next generation.
- The number of available host nests is fixed. And the discovery probability rate by the host bird is considered with the probability \(p_a \in [0,1]\).

For simplicity, the last assumption can be dealt with the fraction \(p_a\) of the \(n\) nests and replaced by new nests (new random solutions), in a nest each egg represents a solution and a new solution represented by a cuckoo egg. The aim is to use and replace new and high fitness solutions (cuckoos) instead of solutions that are not good. This algorithm can be enhanced to the extended form such that each nest has multiple eggs representing a set of solutions\textsuperscript{4,23}. The pseudo code is shown in Figure 1.

When new solution \(x^{(t+1)}\) is generating for \(i\)th cuckoo the following levy flight performed as:
\[
x_i^{(t+1)} = x_i^{(t)} + \partial \oplus \text{levy}(\beta)
\]

Where \(\partial\) is the step size which should be related to the scales of the problem of interest. We can use \(\partial = 1\) in most cases. The product \(\oplus\) means entry wise multiplications\textsuperscript{23}. Levy flight essentially provides a random walk while their random steps are drawn from a levy distribution for large steps.
\[
\text{levy} \sim \mu \sim t^{(-1-\beta)} (0 \leq \beta \leq 2)
\]
4. Differential Evolution Algorithm

Differential Evolution (DE) is one of the evolutionary algorithms that aims at evolving a population of NP D-Dimensional Parameters Value. First phase of this algorithm like most of the other evolutionary algorithms is initialization phase. The initial population should cover the entire search space by uniformly randomizing individuals constrained by lower bound and upper bound parameters $X_{min} = \{x_{min}^1, \ldots, x_{min}^D\}$ and $X_{max} = \{x_{max}^1, \ldots, x_{max}^D\}$. For example the initial value of the jth parameter in the ith individual at the generation $G = 0$ is generated by:

$$x_{i,0}^j = x_{min}^j + rand(0,1). (x_{max}^j - x_{min}^j), \ j = 1, 2, \ldots, D.$$ 

4.1 Mutation Operator

Second phase of the algorithm is mutation operation to produce a mutant vector $V_{iG}$ with respect to each individual $X_{iG}$ in the current population one of mutation strategy is shown as follows:

$$V_{iG} = X_{iG} + F.(X_{r1G} - X_{r2G})$$

The indices $r1, r2, \ldots$, is mutually exclusive integer randomly generated within the range $[1; NP]$, which also different from index $i$, the scaling factor $F$ is a positive control parameter for scaling the difference vector.

4.2 Crossover Operator

The third phase of DE is crossover. It is performed after mutation and is applied to each pair of the target vector $X_{iG}$ and its corresponding mutant vector $V_{iG}$ to generate a trial vector: $(u_{iG}^1, u_{iG}^2, \ldots, u_{iG}^D)$. Crossover can be performed as the following equation:

$$u_{iG}^j = \begin{cases} v_{iG}^j, & \text{if } rand_j[0,1] \le CR \text{ or } j = j_{rand} \\ x_{iG}^j, & \text{otherwise} \end{cases}$$

In equation 11, $i = 1, 2, \ldots D$, $CR$ indicates to crossover rate. It is a constant value within the range $(0;1)$, which controls the fraction of parameter values copied from the mutant vector. $j_{rand}$ is a randomly chosen integer in the range $[1,D]$. As it is shown from Eq 11 if $rand_j[0,1] \le CR$ or $j = j_{rand}$, the binomial crossover operator copies the jth parameter of the mutant vector $V_{iG}$ to the corresponding element in the vector $U_{iG}$.

4.3 Selection

The next generation will be selected among the individuals in current population. The objective function of each trial vector $f(u_{iG})$ is compared with its corresponding target function $f(x_i)$ to decide which individual is superior.
vector $f(x)$. If $f(u) \leq f(x)$ in current population, the trial vector will replace the target vector, so the next generation of population will be created, otherwise the target vector will remain in the population for the next generation. Selection can be performed as follows equation:

$$x_{t+1} = \begin{cases} v_G, & \text{if } f(u_G) \leq f(x_G), \\ x_G, & \text{otherwise} \end{cases}$$

The pseudo code of DE algorithm is shown in Figure 2.

5. Proposed HCSDE Algorithm

In this section we will introduce the proposed hybrid algorithm. As mentioned in section 3, cuckoos don’t make their own nest but they lay their eggs in the nest of other birds called host birds, in this case $p_a$ percent of eggs maybe discovered by the host birds, so they will either throw these alien eggs away or simply abandon their nest and build new nest elsewhere. For simplicity in CS algorithm the fraction $p_a$ percent of the $n$ nests will be replaced by nests that are produced randomly. The randomly production of nests may make tow weaknesses for the CS algorithm. First as the dimension exchange it doesn’t help to have good information exchange among cuckoos, so it causes lower convergence speed. Second it may doesn’t help reaches to better nest and subsequently better fitness.

Differential Evolution (DE) is an evolutionary algorithm proposed by Storn and Price. It aims at evolving a population of NP D-Dimensional Parameters Value. This algorithm as like as most of the Evolutionary algorithms begins with an initial population which represents a set of initial solutions. Then the operators like mutation and crossover and selection are applied to the initial population to produce new population and solutions. This algorithm uses the difference between the selected vectors (individuals and chromosomes). In recent years many proposed algorithms benefits from DE algorithm in their hybrid approaches. Li et al., proposed DE-ABC hybrid algorithm for optimal reactive power flow. They used ABC to enhance the global search of DE algorithm. W. Kwedlo proposed DE-KM Hybrid algorithm. This algorithm uses K-means algorithm to fine-tune each candidate solution obtained by mutation and crossover operators of DE. Sayah et al. proposed DEPSO Hybrid algorithm for no convex economic dispatch problems. In this algorithm they used PSO procedure as an additional mutation operator to improve the global optimization.

In this paper the proposed algorithm benefits from DE operators such as mutation and crossover to produce better solutions instead of random solutions in standard CS algorithm. It also improves the standard CS algorithm in terms of convergence speed and accuracy. The proposed algorithm is named Hybrid of Cuckoo Search (CS) and Differential Evolution (DE) (HCSDE). To have the better local search the proposed algorithm also benefits from Mantegna levy distribution. It is computed as follows:

$$\text{stepsize} = \alpha \frac{1}{10\varphi} \left( \frac{u_j}{v_j} \right)^{1/\alpha} (S - \text{bestnest})$$

Where $u = \varphi \cdot \text{randn} \ [D]$ and $v = \text{randn} \ [D]$. $\text{randn} \ [D]$ is normal distribution of D dimension with mean $= 0$ and $SD=1$ and $\alpha$ indicates the levy step size. The $\varphi$ is computed like Equation (8). $S$ is a randomly selected solution and bestnest is the best solution ever found.

The pseudo code of the HCSDE is shown in Figure 3. The proposed algorithm is much easier to implement in comparison with the hybrid algorithms mentioned above. First step of the algorithm is initializing population. The HCSDE uses Equation (9) to produce initial population. In step 2 CS algorithm uses Mantegna levy distribution to produce new nests based on the Equation (13). It produces $1-p_a$ percent nests of $n$ nests for the algorithm. Then evaluating the fitness of these nests will be done in this step. In step 3 instead of producing $p_a$ percent of nests randomly, the HCSDE benefits from DE algorithm’s operators like mutation and crossover to produce new nests. Then evaluating the produced nests of this step will be done. Finally the HCSDE compares the two obtained fitness values of each step 2 and 3 and it stores the best nest (solution) and best fitness. The HCSDE sorts the
population for each iteration of algorithm. This means that the proposed algorithm feeds step 2 with top nests. After some experiments we found out that it is better to assign these top nests to this step because by implementing this on the proposed algorithm, it reaches better results finally.

Figure 3. The HCSDE pseudo code.

6. Experimental Results

In this section we will describe the results and statistical analysis in detail. In our experiment we employed six datasets named Iris, Wine, Contraceptive Method Choice (denoted as CMC), Wisconsin Breast Cancer (denoted as cancer), Glass and Vowel. They can be found in\textsuperscript{27}. Table 1 shows the properties of these data sets.

6.1 Datasets

Iris data (N = 150, d = 4, k = 3). This is the iris data set. These data sets with 150 random samples of flowers from the iris species Setosa, Versicolor and Virginica are collected by Anderson. From each species there are 50 observations for Sepal Length, Sepal Width, and Petal Length and Petal Width in cm. This dataset is illustrated in Figure 1\textsuperscript{28}.

Wine data (N = 178, d = 13, k = 3). This is the wine data set, from MCI laboratory. These data are the results of chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. There are 178 instances with\textsuperscript{13} numeric attributes in the wine data set. All attributes are continuous. There is no missing attribute value\textsuperscript{28}.

Contraceptive method choice (N = 1473, d = 9, k = 3). CMC is a subset of the 1987 national Indonesia contraceptive prevalence survey. The samples are married women who were either not pregnant or do not know if they were at the time of the interview. The problem is to predict the current contraceptive method choice (like no use, long-term method, or short-term methods) of a woman based on her demographic and socioeconomic characteristics\textsuperscript{28}.

Wisconsin Breast Cancer (N = 683, d = 9, k = 2). The WBC data set consists of 683 objects characterized by nine features: Clump Thickness, Cell Size Uniformity, Cell Shape Uniformity, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nuclei, and Mitoses. There are two categories in the data: malignant (444 objects) and benign (239 objects)\textsuperscript{28}.

Ripleys glass (N = 214, d = 9, k = 6). For which the data were sampled from six different types of glass: building windows float processed (70 objects), building windows non-float processed (76 objects), vehicle windows float processed (17 objects), containers (13 objects), tableware (9 objects), and headlamps (29 objects), each with nine features, which are refractive index, sodium, magnesium, aluminum, silicon, potassium, calcium, barium, and iron\textsuperscript{28}.

| Dataset name | No. of classes | No. of features | Size of dataset (size of classes in parentheses) |
|--------------|----------------|----------------|-----------------------------------------------|
| Iris         | 3              | 4              | 150(50,50,50)                                 |
| Wine         | 3              | 13             | 178(59,71,48)                                 |
| CMC          | 3              | 9              | 1473(629,334,510)                             |
| Cancer       | 2              | 9              | 683(444,239)                                  |
| Glass        | 6              | 9              | 214(70,17,76,13,9,29)                         |
| Vowel        | 6              | 3              | 871(72,89,172,151,207,180)                    |
6.2 Parameters Setting for the Involved Algorithms

In this experiment, each algorithm was run for 50 individual times with random initial solutions on every datasets. The population size of all algorithms was 50. The comparison algorithms are BH, PSO, GSA, and BB-BC by Hatamlou, and also we implemented clustering on original codes of CS algorithm and DE algorithm. After experiments we found out $p_a = 0.7$ and $\beta = 1.9$ have better results for the standard CS and PCR = 0.2 (crossover rate) and $s_c^\text{min} = 0.2$ and $s_c^\text{max} = 0.8$ as scaling factor have better results for DE algorithm. As a measure criterion in this experiment we use the max number of running iterations and the iteration size was set to 500 for every algorithm. All algorithms were implemented in Matlab 2009a using a computer with Intel core 3, 2.5GHz, 4GB RAMS. The operating system of the computer was windows 7.

6.3 Simulation Results

We will discuss the performance of the proposed algorithm, considering the two following criteria:

- Sum of intra-cluster distances as an internal quality measure: The distance between each data vector in a cluster and the centroid of that cluster is calculated and summed up, as defined in Eq. (2).
- Error rate (ER): It is defined as the number of misplaced points over the total number of points in the dataset, as shown in Eq.(14).

\[ \text{Error rate} = \left( \frac{M}{n} \right) \times 100\% \]

Where $n$ denoted to the number of data points and $M$ is the total number of misplaced points.

The summary of the intra-cluster distances obtained by the clustering algorithms is shown in Table 2. These

| Dataset | HCSDE | CS | DE | PSO | GSA | BB-BC | BH |
|---------|-------|----|----|-----|-----|-------|----|
| Iris    | Best  | 96.6554 | 96.6723 | 96.6559 | 96.67395 | 96.678794 | 96.67648 | 96.65589 |
|         | Average | 96.6554 | 96.7122 | 96.6751 | 98.14236 | 96.73105 | 96.76537 | 96.65681 |
|         | Worst  | 96.6554 | 96.8814 | 96.6897 | 99.76952 | 96.82463 | 97.42865 | 96.66306 |
|         | Std    | 0.000005 | 0.2236 | 0.0321 | 0.84207 | 0.02761 | 0.20456 | 0.00173 |
| Wine    | Best  | 16292.2173 | 16293.9685 | 16292.8548 | 16,304.48576 | 16,313.87620 | 16,298.67356 | 16,293.41995 |
|         | Average | 16292.4854 | 16294.3209 | 16293.8971 | 16,316.27450 | 16,374.30912 | 16,303.41207 | 16,303.92076 |
|         | Worst  | 16292.8825 | 16294.8278 | 16294.3317 | 16,342.78109 | 16,428.86494 | 16,310.11354 | 16,300.22613 |
|         | Std    | 0.2472 | 0.8252 | 1.4220 | 12.60275 | 34.67122 | 2.66198 | 1.65127 |
| CMC     | Best  | 5532.2265 | 5556.996 | 5537.7896 | 5539.17452 | 5542.27631 | 5534.09483 | 5532.88323 |
|         | Average | 5532.7369 | 5561.4187 | 5538.5826 | 5547.89320 | 5581.94502 | 5574.75174 | 5553.63122 |
|         | Worst  | 5532.9531 | 5567.1565 | 5539.1864 | 5561.6492 | 5658.76293 | 5644.70264 | 5534.77378 |
|         | Std    | 0.1632 | 3.6589 | 1.5364 | 7.35617 | 41.13648 | 39.43494 | 0.59940 |
| Cancer  | Best  | 2964.3869 | 2965.1528 | 2964.4108 | 2974.48092 | 2965.76394 | 2964.38753 | 2964.38878 |
|         | Average | 2964.3869 | 2965.6918 | 2964.6059 | 2981.78653 | 2972.66312 | 2964.38798 | 2964.39539 |
|         | Worst  | 2964.3869 | 2967.0782 | 2965.1127 | 3053.49132 | 2993.24458 | 2964.38902 | 2964.45074 |
|         | Std    | 0.0003 | 0.8652 | 0.8325 | 10.43651 | 8.91860 | 0.00048 | 0.00921 |
| Glass   | Best  | 212.2359 | 220.1258 | 213.2861 | 223.90546 | 224.98410 | 223.89410 | 210.51549 |
|         | Average | 215.7445 | 225.1982 | 218.9863 | 230.49328 | 233.54329 | 231.23058 | 211.49860 |
|         | Worst  | 225.0188 | 227.0223 | 231.2943 | 246.08915 | 248.36721 | 243.20883 | 213.95689 |
|         | Std    | 3.54367 | 5.6623 | 7.1493 | 4.79320 | 6.13946 | 4.65013 | 1.18230 |
| Vowel   | Best  | 149867.64795 | 149417.318 | 149181.6447 | 152,461.56473 | 151,317.56392 | 149,038.51683 | 148,958.61373 |
|         | Average | 149479.8741 | 150186.1283 | 149966.6667 | 153,218.23418 | 152,931.81044 | 151,010.03392 | 149,848.18144 |
|         | Worst  | 150239.5932 | 150841.4059 | 150927.4294 | 158,987.08231 | 155,346.69521 | 153,090.44077 | 153,058.98663 |
|         | Std    | 596.17303 | 1576.3697 | 1423.1212 | 2945.23167 | 2486.70285 | 1859.32353 | 1306.95375 |
values arranged in best, average, worst and STD values. Best indicates the best total within-cluster variance for 50 runs. Average indicates the mean total within-cluster variance for 50 runs. Worst indicates to the worst total within cluster variance for 50 runs and STD indicates the Standard Deviation. It can be seen from Table 2 and 3 that HCSDE outperforms other algorithms in all datasets except in glass dataset.

In iris dataset HCSDE and BH have results close to each other but standard deviation of HCSDE is better than the other algorithms which mean that HCSDE reaches minimum value each time. PSO is a little worse than the other algorithms. The convergence speed of HCSDE is much better than the others. The CS algorithm has the worst convergence speed; this is shown in Figure 4(a).

In wine dataset HCSDE outperforms other algorithms and has better intra-cluster distance value. It also has efficient standard deviation in comparison with the other algorithms. DE has closer result to the HCSDE, PSO, GSA and BB-BC algorithms in worst case. The convergence

Table 3. Error rate for the employed six datasets

| Datasets | HCSDE(%) | DE(%) | CS(%) | PSO(%) | GSA(%) | BB-BC(%) | BH(%) |
|----------|----------|-------|-------|--------|--------|----------|-------|
| Iris     | 10.02    | 10.05 | 10.05 | 10.06  | 10.04  | 10.05    | 10.02 |
| Wine     | 28.39    | 28.51 | 28.53 | 28.79  | 29.15  | 28.52    | 28.47 |
| CMC      | 54.32    | 54.51 | 54.73 | 54.50  | 55.67  | 54.52    | 54.39 |
| Cancer   | 3.52     | 3.71  | 3.72  | 3.79   | 3.74   | 3.70     | 3.70  |
| Glass    | 36.82    | 41.63 | 41.89 | 41.20  | 41.39  | 41.37    | 36.51 |
| Vowel    | 41.62    | 42.25 | 42.32 | 42.39  | 42.26  | 41.89    | 41.65 |

Figure 4. Convergence speed of the employed datasets for the best minimum total within-cluster variance of HCSDE, CS, DE, PSO, GSA, BB-BC and BH. (a) Iris dataset. (b) Wine dataset. (c) CMC dataset. (d) Cancer dataset.
speed of HCSDE is better than the other algorithms. This is shown in Figure 4(b).

In CMC dataset, HCSDE can produces much better results in terms of convergence speed, accuracy and standard deviation in comparison with the other algorithms. BH can produces results close to the proposed algorithm and the results of the other algorithm are much weaker than the two above algorithms. The convergence speed of the algorithms is shown in Figure 4(c).

In cancer dataset HCSDE, DE, BB-BC and BH have closer results to each other but the proposed algorithm can produces better results finally. CS, GSA and PSO have weaker results in comparison with three above algorithms. The proposed algorithm also has superiority in obtaining better Standard Deviation and convergence speed. These are shown in Table 2 and Figure 4(d).

### Table 4. The best centroid obtained by HCSDE for iris dataset

| Center1 | Center2 | Center3 |
|---------|---------|---------|
| 5.01213 | 5.93432 | 6.73334 |
| 3.40310 | 2.79779 | 3.06785 |
| 1.47163 | 4.41789 | 5.63007 |
| 0.23540 | 1.41726 | 2.10679 |

### Table 5. The best centroid obtained by HCSDE for wine dataset

| Center1 | Center2 | Center3 |
|---------|---------|---------|
| 12.84199 | 12.53398 | 13.76041 |
| 2.53057 | 2.30606 | 1.90225 |
| 2.38489 | 2.29479 | 2.51417 |
| 19.53300 | 21.32272 | 16.96509 |
| 98.94535 | 92.54145 | 105.26074 |
| 2.07838 | 2.03227 | 2.85407 |
| 1.48163 | 1.62664 | 3.02801 |
| 0.43163 | 0.41913 | 0.29086 |
| 1.39694 | 1.45703 | 1.97766 |
| 5.78721 | 4.36012 | 5.64486 |
| 0.93370 | 0.96000 | 1.06261 |
| 2.20620 | 2.46834 | 3.04350 |
| 686.97084 | 463.62479 | 1137.34940 |

### Table 6. The best centroid obtained by HCSDE for CMC dataset

| Center1 | Center2 | Center3 |
|---------|---------|---------|
| 24.42163 | 43.63550 | 33.49849 |
| 3.03939 | 3.01171 | 3.13096 |
| 3.50907 | 3.44733 | 3.55849 |
| 1.79540 | 4.58173 | 3.64229 |
| 0.92988 | 0.79866 | 0.79782 |
| 0.79057 | 0.76066 | 0.69882 |
| 2.29378 | 1.82831 | 2.09315 |
| 2.97597 | 3.43843 | 3.28428 |
| 0.03555 | 0.07985 | 0.05632 |

### Table 7. The best centroid obtained by HCSDE for cancer dataset

| Center1 | Center2 |
|---------|---------|
| 2.88935 | 7.11709 |
| 1.12775 | 6.64108 |
| 1.20661 | 6.62546 |
| 1.16411 | 5.61446 |
| 1.99348 | 5.24086 |
| 1.12129 | 8.10109 |
| 2.00549 | 6.07992 |
| 1.10130 | 6.02189 |
| 1.03157 | 2.32554 |

### Table 8. The best centroid obtained by HCSDE for glass dataset

| Center1 | Center2 | Center3 | Center4 | Center5 | Center6 |
|---------|---------|---------|---------|---------|---------|
| 1.52040 | 1.51870 | 1.51877 | 1.51614 | 1.52706 | 1.51873 |
| 14.62317 | 13.79038 | 13.00667 | 13.9656 | 12.01216 | 13.79624 |
| 0.07278 | 0.47184 | 0.00034 | 3.52343 | 0.02697 | 3.54930 |
| 2.21017 | 1.51355 | 3.01362 | 1.37150 | 1.11698 | 0.96008 |
| 73.26098 | 72.86623 | 70.58047 | 72.83990 | 72.02272 | 71.86814 |
| 0.045743 | 0 | 6.21000 | 0.58272 | 0.21703 | 0.17177 |
| 8.66474 | 11.11952 | 6.94609 | 8.36816 | 14.30366 | 9.50148 |
| 1.03802 | 0 | 0.00074 | 0.00519 | 0.18216 | 0.03127 |
| 0.01014 | 0 | 0.00217 | 0.05280 | 0.09212 | 0.05632 |

### Table 9. The best centroid obtained by HCSDE for vowel dataset

| Center1 | Center2 | Center3 | Center4 | Center5 | Center6 |
|---------|---------|---------|---------|---------|---------|
| 507.08502 | 623.61847 | 375.49177 | 438.88982 | 357.24573 | 407.78329 |
| 1840.10251 | 1309.64404 | 2149.65654 | 987.55401 | 2291.91169 | 1017.83509 |
| 2556.73888 | 2333.67375 | 2678.93173 | 2665.36544 | 2977.39386 | 2317.72089 |
In Glass dataset, BH algorithm outperforms the other algorithms. In this dataset the proposed algorithm is weaker than the BH algorithm but its result is really close to the BH algorithm and also is much better than the other five algorithms. GSA algorithm has the worst result in this dataset with respect to the other algorithms.

In vowel dataset the proposed algorithm has superiority with respect to the other six algorithms. It can obtain much better accuracy and also has reasonable standard deviation. BH algorithm has closer result to the proposed algorithm and PSO algorithm has the worst results in comparison with the other algorithms.

The best centroids obtained by the proposed algorithm are shown in Tables 4-9. The values of these tables can be used for reaching to the best results of Table 2 by assigning the data objects within each dataset to the corresponding centroids in Tables 4-9. Error rate obtained by the clustering algorithms from 50 simulations on the employed datasets is shown in Table 3. In five datasets, the proposed algorithm has minimum error rate except in glass dataset. In iris dataset HCSDE and BH have the minimum error rate except in glass dataset. In iris dataset HCSDE and BH have the minimum error rate in comparison with the other algorithms. In wine, CMC, cancer and vowel HCSDE also has superiority to the others in obtaining the minimum error rate. In glass dataset BH obtained the minimum error rate, however there is no correlation between the intra-cluster distance and the error rate but HCSDE is weaker than BH in this dataset as it was for intra-cluster distance value too.

6.4 Statistical Analysis
In this subsection we will have a statistical test on the proposed algorithm.

6.4.1 Wilcoxon Signed Rank Test
Swarm intelligence-based algorithms have stochastic nature. Because of this point they sometimes reach the solutions by chance, so it is more important to use statistical tools to test the problem-solving success. A problem based pair wise test is used to check which one of the two algorithms solves a specific numerical optimization problem with greater statistical success. In this paper we also used the global minimum values obtained by 50 runs for the problem-based pair wise of the algorithm. We employed Wilcoxon Signed Rank Test for pair wise comparison, with the statistical significance value 0.05. Considering the two algorithms named x and y, the null hypothesis $H_0$ for this test, shows there is no difference between the median of the solution obtained by the algorithm x and the median of the solution obtained by the algorithm y. To determine which algorithm obtained better results, the size of ranks were provided by the Wilcoxon on signed rank test. In Table 10, $T_+$ is the sum of ranks for the problem in which the first algorithm outperforms the second one (its rival) and $T_-$ shows the sum of the ranks for the second algorithm (the proposed algorithm here) in which it has superiority. “+” indicates the cases in which null hypothesis was rejected and in fact HCSDE shows its superiority in the problem-based comparison test at the 95% significant level ($\alpha = 0.05$), “.” indicates the cases in which the null hypothesis was rejected and HCSDE does not have the superiority, “=” indicate to the cases which there is no statistical difference between the two algorithms. It is shows from Table 10 that HCSDE is statistically more successful in comparison with the other algorithms.

7. Conclusions and Future Works
In this paper we proposed an efficient hybrid algorithm for data clustering. The proposed algorithm benefits from DE to produce new nests in standard CS in order to obtain better results and it also benefits from Mantegna Levy Flight to boost the local search. It shows that the proposed algorithm outperforms the others and has better total within-cluster variance value in comparison with CS, DE, PSO, GSA, BB-BC and BH. In section 6 we showed that the proposed algorithm satisfies the characteristics that we claimed in section 1. We tested HCSDE with six well known real datasets from UCI machine learning repository. It is obvious from Table 2 and 3 that the proposed algorithm has better accuracy and error rate with respect to the other algorithms. However the proposed algorithms get into local minima in some cases, so improving features of the algorithm could be considered as the future work.
Table 10. Determining the algorithms that statistically provide the best solution using Wilcoxon Signed-Rank Test

| Datasets | CS vs. HCSDE | DE vs. HCSDE | PSO vs. HCSDE | GSA vs. HCSDE | BB-BC vs. HCSDE | BH vs. HCSDE |
|----------|--------------|--------------|---------------|---------------|----------------|--------------|
| Iris     | P-value      | 4.544e-101   | 1.764e-045    | 2.173e-092    | 1.713e-075     | 2.490e-074   | 2.489e-045   |
|          | T+           | 8            | 58            | 6             | 12             | 0            | 35           |
|          | T-           | 492          | 442           | 494           | 488            | 500          | 465          |
| Winner   | +            | +            | +             |              | +              | +            |
| Wine     | P-value      | 5.975e-033   | 1.756e-005    | 7.259e-081    | 5.059e-085     | 1.535e-065   | 1.510e-024   |
|          | T+           | 6            | 59            | 0             | 11             | 50           | 20           |
|          | T-           | 494          | 441           | 500           | 489            | 450          | 480          |
| Winner   | +            | +            | +             | +             | +              | +            |
| CMC      | P-value      | 5.363e-050   | 4.881e-014    | 4.131e-022    | 2.274e-026     | 5.194e-019   | 2.968e-008   |
|          | T+           | 29           | 12            | 11            | 2              | 10           | 85           |
|          | T-           | 471          | 488           | 489           | 498            | 490          | 415          |
| Winner   | +            | +            | +             | +             | +              | +            |
| Cancer   | P-value      | 1.256e-056   | 9.427e-011    | 7.191e-071    | 4.525e-082     | 1.211e-025   | 3.842e-026   |
|          | T+           | 24           | 13            | 4             | 13             | 20           | 143          |
|          | T-           | 476          | 487           | 496           | 487            | 480          | 357          |
| Winner   | +            | +            | +             | +             | +              | +            |
| Glass    | P-value      | 1.697e-064   | 2.146e-038    | 2.206e-053    | 9.973e-052     | 3.6693e-050  | 0.2664       |
|          | T+           | 0            | 5             | 6             | 0              | 6            | 327          |
|          | T-           | 500          | 495           | 494           | 500            | 494          | 173          |
| Winner   | +            | +            | +             | +             | +              | +            |
| Vowel    | P-value      | 1.192e-048   | 2.2101e-032   | 2.017e-056    | 6.247e-039      | 0.2620       | 0.0052       |
|          | T+           | 37           | 54            | 65            | 61             | 95           | 182          |
|          | T-           | 463          | 446           | 435           | 439            | 405          | 318          |
| Winner   | +            | +            | +             | +             | +              | +            |

8. References

1. Karaboga D, Ozturk C. A novel clustering approach. Artificial Bee Colony (ABC) algorithm. Appl Soft Comput. 2011; 11:652–57.
2. Kennedy J, Eberhart R. Particle swarm optimization. IEEE International Joint Conference on Neural Network; 1995. p. 1942–48.
3. Runkler TA. Ant colony optimization of clustering models. Research Articles. Int J Intell Syst. 2005; 20:1233–51.
4. Yang XS, Deb S. Cuckoo search via levy flight. Proc of World Congress on Nature and Biologically Inspired Computing; India. 2009 Dec. p. 210–4.
5. Storn R, Price K. Differential evolutional simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization. 1997; 11(4):341–59.
6. Yao H, et al. An improved K-means clustering algorithm for fish image segmentation. Mathematical and Computer Modeling. 2013; 58(3):790–8.
7. Hilas CS, Mastorocostas PA. An application of supervised and unsupervised learning approaches to telecommunications fraud detection. Knowledge-Based Systems. 2008; 21(7):721–6.
8. Fodeh SJ, et al. Complementary ensemble clustering of biomedical data. Journal of Biomedical Informatics. 2013; 46(3):436–43.
9. Liu CM, Lee C-H, Wang L-C. Distributed clustering algorithms for data-gathering in wireless mobile sensor networks. Journal of Parallel and Distributed Computing. 2007; 67(11):1187–200.
10. Jain AK. Data clustering: 50 years beyond K-means. Pattern Recognition Letter. 2010; 31:651–66.
11. Selim SZ, Ismail MA. K-means-type algorithms. A generalized convergence theorem and characterization of local optimality. IEEE Trans Pattern Anal Mach Intell. 1984; 6:81–7.
12. Zalik KR. An efficient k-means clustering algorithm. Pattern Recognition Letters. 2008; 29:1385–91.
13. Niknam T, Olamaie J, Amiri B. A hybrid evolutionary algorithm based on ACO and SA for cluster analysis. Journal of Applied Science. 2008; 8(15):2695–702.
14. Niknam T, Amiri B, Olamaie J, Arefi A. An efficient hybrid evolutionary optimization algorithm based on PSO and SA for clustering. Journal of Zhejiang University Science A. 2008; 10(4):512–9. DOI:10.1631/jzus.A0820196.
15. Niknam T, Amiri B. An efficient hybrid approach based on PSO, ACO and K-means for cluster analysis. Applied Soft Computing. 2010; 10(1):183–97.
16. Tsai CY, Kao I-W. Particle swarm optimization with selective particle regeneration for data clustering. Expert System with Application. 201; 38:6565–76.
17. Shelokar PS, Jayaraman VK, Kulkarni BD. An ant colony approach for clustering. Anal Chim Acta. 2004; 509(2):187–95.
18. Paterlini S, Krink T. Differential evolution and particle swarm optimization in partitional clustering. Computational statistics and Data Analysis. 2006; 50:1220–47.
19. Hatamlou A, Abdullah S, Nezamabadi-Pour H. A combined approach for clustering based on k-means and gravitational search algorithms. Swarm and Evolutionary Computation. 2012; 6:47–52.
20. Yan X, Zhu Y, Zou W, Wang L. A new approach for data clustering using hybrid artificial bee colony algorithm. Neurocomputing. 2012; 97:241–50.
21. Krishna K, Murty MN. Genetic K-means algorithm. IEEE Transaction of System Man Cybernetics Part B-Cybernetics. 1999; 29:433–39.
22. Yang XS. Nature-inspired metaheuristic algorithms. 2nd Edition. Luniver Press; 2010.
23. Yang XS, Deb S. Engineering optimization by cuckoo search. Int J Math Modelling and Num Optimization. 2010; 4:330–43.
24. Li Y, Wang Y, Li B. A hybrid artificial bee colony assisted differential evolution algorithm for optimal reactive power flow. International Journal of Electrical Power and Energy Systems. 2013; 52:25–33.
25. Kwedlo W. A clustering method combining differential evolution with the k-means algorithm. Pattern Recognition Letters. 2011; 32(12):1613–21.
26. Sayah S, Hamouda A. A hybrid differential evolution algorithm based on particle swarm optimization for non convex economic dispatch problems. Applied Soft Computing. 2013; 13(4): 1608–19.
27. Blake CL, Merz CJ. UCI repository of machine learning databases. Available from: http://archive.ics.uci.edu/ml/datasets.html
28. Niknam T, Firouzi BB, Nayeripour M. An efficient hybrid evolutionary algorithm for cluster analysis. World Applied Sciences Journal. 2008; 4(2):300–7.
29. Hatamlou A. Black hole: A new heuristic optimization approach for data clustering. Information Sciences. 2013; 222:175–84.
30. Derrac J, Garcia S, Molina D, Herrera F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation. 2011; 1:318.