

## Evaluation of Differential Evolution Algorithm with Various Mutation Strategies for Clustering Problems

Pyae Pyae Win Cho<sup>1</sup>, Thi Thi Soe Nyunt<sup>2</sup>

<sup>1,2</sup>University of Computer Studies, Yangon

---

### Article Info

#### Article historys:

Received Feb 11, 2020

Revised Dec 15, 2020

Accepted Dec 21, 2020

---

#### Keywords:

Differential Evolution

Mutation Strategy

Clustering

---

### ABSTRACT

Evolutionary Algorithms (EAs) based pattern recognition has emerged as an alternative solution to data analysis problems to enhance the efficiency and accuracy of mining processes. Differential Evolution (DE) is one rival and powerful instance of EAs, and DE has been successfully used for cluster analysis in recent years. Mutation strategy, one of the main processes of DE, uses scaled differences of individuals that are chosen randomly from the population to generate a mutant (trial) vector. The achievement of the DE algorithm for solving optimization problems highly relies on an adopted mutation strategy. In this paper, an empirical study was presented to investigate the effectiveness of six frequently used mutation strategies for solving clustering problems. The experimental tests were conducted on the most widely used data set for EAs based clustering, and the quality of cluster solutions and convergence characteristics of DE variants were evaluated. The obtained results pointed out that the mutation strategies that use the guidance information from the best solution manage to find more stable results whereas the random mutation strategies are able to find high quality solutions with slower convergence rate. This study aims to provide some information and insights to develop better DE mutation schemes for clustering.

Copyright © 2020 Institute of Advanced Engineering and Science.  
All rights reserved.

---

### Corresponding Author:

Pyae Pyae Win Cho,

University of Computer Studies, Yangon,

No. (4) Main Road, Shwe Pyi Thar Township, Yangon, Myanmar.

Email: pyaepyaewincho@ucsy.edu.mm

---

## 1. INTRODUCTION

Evolutionary algorithms (EAs), motivated by Darwin's theory of natural selection, have become a powerful way to solve several different optimization problems in various domains [1]. Among them, the Differential Evolution (DE) algorithm is a simple, competent, and robust stochastic search strategy based on population, and it has been successfully used to catch global optimum on high-dimensional continuous problems [2]. Like a standard EA, DE applies evolution processes such as mutation, crossover and selection to transfer from one generation to the next. It differs significantly from other EAs in the fact that the distance between pairs of randomly selected individuals is used to modify the solution, and the selected solution's position guides the track of the search process. To employ the mutation operator in DE, there exist a few different mutation strategies that determine the solution to be modified and the number of different vectors to be used to find the distance for modification [3, 4]. The effectiveness of DE heavily depends on the chosen mutation strategy due to different mutation strategies that can guide different tracks toward exploration and exploitation.

In recent years, DE has been widely utilized to solve clustering problems due to its ability to enhance the solution quality. It has been widely used to perform clustering independently [5-7] or incorporate it into the existing clustering approaches [8-10]. Paterlini and Krink described an innovative approach for DE based clustering [5, 6]. They studied the performance superiority between genetic algorithm (GA), particle swarm optimization (PSO), and DE, and concluded that DE is more suitable for cluster analysis. Some paper proposed the combination of DE with local search approaches to achieve considerably better efficiency [8-10].

Nevertheless, it is needed that the comparisons on clustering performance of different mutation strategies for cluster analysis. In this paper, an empirical analysis is presented to compare and examine the performance of DE with different variants of mutation strategy for clustering problems. It is expected that the acquired information insight from the experiments may be useful and helpful to employ optimal mutation strategy for future DE researches in the clustering domain.

In the next section of the paper, a brief explanation of a traditional DE algorithm and different mutation strategies used in DE are presented. The DE based clustering method is explained in Section 3. In the fourth part, the outcomes of experimental tests are shown, and in Section 5, the paper is completed with a conclusion.

## 2. BACKGROUND

In this section, the basic structure of DE algorithm is firstly described, and then different variants of the mutation strategy used in DE are briefly explained.

### 2.1. Differential Evolution Algorithm

DE is an innovative heuristic population-based search approach that had been proposed by Storn and Price in 1995. It has become one of the most successful and widely used EAs to solve several real-world continuous global optimization problems in various domains [2, 3]. Like a standard EA, DE maintains a population of individuals that are a sample of candidate solutions to an optimization problem. Hence, an initial population is created through random sampling with uniform distribution at the beginning of the algorithm. And then, DE iteratively performs three consecutive steps (namely mutation, crossover, and selection) until a stopping situation is reached.

Let  $X_{i,g} = \{x_{i,g}^1, x_{i,g}^2, \dots, x_{i,g}^d\}$  is the  $i^{\text{th}}$  solution (individual) of the population,  $P = \{X_1, X_2, \dots, X_{NP}\}$  at the  $g^{\text{th}}$  iteration where  $d$  is the data dimensionality and  $NP$  is the size of population.

#### 2.1.1. Mutation

A trial vector  $V_{i,g}$  is generated for each parent solution  $X_{i,g}$  by perturbing a target solution,  $X_{i1,g}$  with a scaled difference as follows:

$$V_{i,g} = X_{i1,g} + f(X_{i2,g} - X_{i3,g}) \quad (1)$$

Where  $i$  is an integer within  $[1, NP]$ ,  $i_1$ ,  $i_2$ , and  $i_3$  are random integers within  $[1, NP]$  such that  $i \neq i_1 \neq i_2 \neq i_3$ , and then  $f$  is a scaled factor within  $(0, \infty)$ .

#### 2.1.2 Crossover

At the crossover phase, an offspring vector,  $U_{i,g}$  is usually generated by applying binomial crossover operator as follows:

$$u_{i,g}^j = \begin{cases} v_{i,g}^j & \text{if } \text{rand}(j) \leq CR \\ x_{i,g}^j & \text{otherwise} \end{cases} \quad (2)$$

Where  $i$  an integers within  $[1, NP]$ ,  $j$  is an integer within  $[1, d]$ ,  $\text{rand}(j) \in U(0,1)$  and the crossover rate,  $CR \in (0,1)$ .

#### 2.1.3 Selection

In the selection phase, the parent solution in the current population and its offspring vector are compared to determine which will remain in the next generation (iteration). The fitter solution is selected and added to the new population. For the maximizing problem, the solution vector for the next iteration is chosen according to the following;

$$X_{i,g+1} = \begin{cases} U_{i,g} & \text{if } f(U_{i,g}) > f(X_{i,g}) \\ X_{i,g} & \text{otherwise} \end{cases} \quad (3)$$

Where  $f(U_{i,g})$  indicates the fitness value of offspring and  $f(X_{i,g})$  denotes the fitness value of  $i^{\text{th}}$  parent in the current population.

## 2.2. Different Mutation Strategies in Differential Evolution

Recent research works proposed numerous variants to the basic DE. In the literature, a notation of DE/x/y/z is commonly used to categorize these different variants [1]. In this notation, x indicates the way of choosing a target solution, y specifies the number of pairs of difference vectors applied, and the last symbol, z, identifies the adopted crossover operator. This paper intends to describe various mutation strategies. Thus, the notation DE/x/y is applied, and the character z is omitted. The random mutation strategy, DE/rand/1 is typically used in a standard DE algorithm. The mutation strategies [2, 3] that are frequently used are as follows.

### 2.2.1. Random Mutation Strategy

DE/rand/1 and DE/rand/2 are the random mutation strategies that use one difference vector and two difference vectors, respectively. As mentioned in above, DE/rand/1 creates the trial vector with three randomly chosen solution vectors while DE/rand/2 uses five randomly selected solution vectors to generate the trial vector according to the following equation;

$$V_{i,g} = X_{i1,g} + f_1(X_{i2,g} - X_{i3,g}) + f_2(X_{i4,g} - X_{i5,g}) \quad (4)$$

Where  $f_1$  and  $f_2$  are two control parameters to scale differences of vectors,  $i1$ ,  $i2$ ,  $i3$ ,  $i4$ , and  $i5$  are disjoint randomly generated integers within  $[1, NP]$ .

### 2.2.2. Best Mutation Strategy

The best mutation strategy applies the fittest solution vector in the population as the target vector. DE/best/1 and DE/best/2 represent two types of the best mutation strategy that use one difference vector and two difference vectors, respectively. These strategies generate the trial vectors as follows;

$$V_{i,g} = X_{best,g} + f(X_{i1,g} - X_{i2,g}) \quad (5)$$

$$V_{i,g} = X_{best,g} + f_1(X_{i1,g} - X_{i2,g}) + f_2(X_{i3,g} - X_{i4,g}) \quad (6)$$

Where  $f$ ,  $f_1$  and  $f_2$  are the scaling factors within  $(0, \infty)$ , and  $i1$ ,  $i2$ ,  $i3$ , and  $i4$  are disjoint randomly generated integers within  $[1, NP]$ .

### 2.2.3. Current to Random Mutation Strategy

The notation DE/current-to-rand/1 indicates the current to random mutation strategy. This strategy uses a parent solution as a target vector and employed two difference vectors to produce a trial vector. The first difference is the difference between one random solution and the parent solution, whereas the rest is computed from two randomly selected vectors. DE/current-to-rand/1 produces the trial vector according to the following equation;

$$V_{i,g} = X_{i,g} + f_1(X_{i1,g} - X_{i,g}) + f_2(X_{i2,g} - X_{i3,g}) \quad (7)$$

Where  $f_1$  and  $f_2 \in (0, \infty)$  are the scaling factors, and  $i1$ ,  $i2$  and  $i3$  are different randomly generated indexes within  $[1, NP]$ .

### 2.2.4. Current to Best Mutation Strategy

This strategy is also known as the target to best mutation strategy and represented by the notation DE/current-to-best/1. It uses the parent solution as a target vector and applies two difference vectors to mutate the target vector. The first difference vector is calculated from the best and parent solutions, whereas the rest is computed from two randomly selected vectors. The trial vector is produced as followings;

$$V_{i,g} = X_{i,g} + f_1(X_{best,g} - X_{i,g}) + f_2(X_{i1,g} - X_{i2,g}) \quad (8)$$

Where  $f_1$  and  $f_2 \in (0, \infty)$  are the scale number for controlling difference vector, and  $i1$  and  $i2 \in [1, NP]$  that are different randomly generated indexes.

## 3. DIFFERENTIAL EVOLUTION BASED CLUSTERING ALGORITHM

DE maintains a number of possible solutions to the problem as a population. Each possible solution is encoded as a chromosome (individual). For applying DE to solve clustering problems, a cluster solution for the given data set is encoded as an individual. And then, cluster validity measures are used as objective functions to evaluate the fitness of the solution [11].

In this paper, centroid-based representation is used where a chromosome is encoded by real numbers, which represents the coordinates of centroids of a cluster solution. If a chromosome encodes  $k$  clusters of a  $d$ -dimensional dataset,  $k*d$  is the length of this chromosome. Each chromosome of the initial population is constructed as follows;  $X_i = \{x_i^1, x_i^2, \dots, x_i^d, x_i^{d+1}, x_i^{d+2}, \dots, x_i^{2d}, \dots, x_i^{(k-1)d+1}, x_i^{(k-1)d+2}, \dots, x_i^{kd}\}$  where the very first  $d$ -dimensional vector stands for the first cluster centroid, the second  $d$ -dimensional vector denotes the coordinate of the second cluster center, and the last  $d$ -dimensional vector represent the  $k^{\text{th}}$  cluster centroid for the given data set. The total intra-cluster distance [7] is used as an objective function to compute the fitness of each chromosome.

$$TotalIntraClusterDistance = \sum_{j=1}^k \sum_{d \in C_j} Dist(d, c_j) \quad (9)$$

Where  $k$  is the number of cluster,  $C_j$  is the  $j^{\text{th}}$  cluster,  $d$  is a data point in  $C_j$ ,  $c_j$  is the center of  $C_j$ , and  $Dist$  is the Euclidean distance [12] between data point  $d$  and the center  $c_j$  of the cluster  $C_j$ .

In DE based clustering algorithm, each chromosome is initialized with  $k$  randomly selected cluster centers from a given dataset to construct an initial population. To compute the fitness of each chromosome, Euclidean distance between each data point and all cluster center of the chromosome is firstly calculated, and then the data points are assigned to the closet cluster, and finally, the sum of intra-cluster distance of each

cluster is calculated. The population for the next generation is produced by mutation, crossover and selection. The best solution of the final population is the optimal cluster solution for the given dataset. The process of DE based clustering algorithm is given in Algorithm 1.

---

**Algorithm 1.** Differential Evolution based Clustering Algorithm (DE-C)

---

- 1: Input: Dataset (D), Number of cluster (k), Maximum iteration(Itr), Number of population (NP), Scaled factor (f), Crossover rate (CR)
  - 2: Output: Cluster solution
  - 3: Generate each chromosome by selecting k data points from D
  - 4: For each chromosome do
  - 5:   For each data point p
  - 6:     Compute the Euclidean distance between data point p and all of the cluster centers
  - 7:     Assign the data point to the closet cluster
  - 8:     Compute the fitness of the chromosome according to eq. (9)
  - 9:   End.
  - 10: End.
  - 11: While the number of iteration is not equal to Itr do
  - 12:   Create a trial vector by applying the mutation operation
  - 13:   Create an offspring by applying the binomial crossover operator
  - 14:   Compute the fitness of the offspring
  - 15:   Update the population by evaluating the fitness of the parent and offspring vectors based on the selection operation
  - 16: End.
- 

#### 4. EXPERIMENTAL STUDY

The main aim of this work is to provide some valuable information for developing a simple, efficient and robust DE based clustering algorithm. The most well-known, simple and efficient mutation strategies were taken into account in this work. The clustering performance of DE algorithms with six different mutation strategies is tested on some real datasets from the UCI machine learning repository [13]. Seven UCI standard datasets that are frequently used for metaheuristics-based clustering [14] are utilized. The summary of these datasets is shown in Table 1.

##### 4.1. Experimental Setup

For all DE based clustering algorithms with different mutation strategies, the crossover rate and the size of the population were respectively set to 0.9 and 100 [7, 15], and the scaling factor was set as follows:  $f=0.5$ ,  $f_1=0.3$  and  $f_2=0.3$ . The number of maximum iteration was set to 100. The initial population was constructed in a similar fashion such that each chromosome was composed of k distinct data points that were randomly selected from the dataset. The algorithms were implemented in java programming language on Intel Core i7 processor, 8GB memory, and 64-bit operating system. Each algorithm was executed 30 times independently for each dataset.

Table 1. Summary of the used datasets

Datasets	No. of Attributes	No. of Instances	No. of Cluster
Iris (Iris plant data)	4	150	3
Wine (Wine recognition data)	13	178	3
Thyroid (Thyroid Disease Data)	5	215	3
Breast Cancer (Wisconsin Diagnostic Breast Cancer Data)	9	683	2
Pima (Pima Indians Diabetes Data)	18	768	2
Glass (Glass Identification Data)	10	214	6
Ecoli (Protein Localization Sites)	8	336	8

The quality of obtained cluster solutions and convergence speed of different DE variants were compared. The quality of clustering solutions was compared according to the following criteria:

- The objective function values (total intra-cluster distance defined in eq. (9) )
- Sum of squared error (SSE) [12]: It calculate the sum of the squared distances from each data point in a cluster to the center of this cluster as follows:

$$SSE = \sum_{j=1}^k \sum_{d \in C_j} Dist(d, c_j)^2 \tag{10}$$

Where k represents the number of clusters,  $C_j$  stands for the jth cluster, d is a data point in  $C_j$ ,  $c_j$  is the center of  $C_j$ , and  $Dist$  is the Euclidean distance between data point d and the center  $c_j$  of the cluster  $C_j$ . Minimum SSE indicates better cluster solutions.

Table 2. Comparison of objective function values

Dataset	Mutation Strategy	Worst	Best	Mean	Std
Iris	DE/rand/1	97.91498	96.91073	97.3809244	0.286711212
	DE/rand/2	104.59639	100.58503	102.1337669	1.18710493
	DE/best/1	97.66495	96.70644	97.1257265	0.287769934
	DE/best/2	96.688	96.6557	96.6670415	0.010426543
	DE/current-to-rand/1	98.56196	97.7142	98.2257235	0.2330539
Wine	DE/current-to-best/1	96.85357	96.676285	96.7280826	0.067527291
	DE/rand/1	16296.588	16291.879	16293.9099	1.798194616
	DE/rand/2	16424.855	16352.439	16386.8182	24.68133014
	DE/best/1	16324.859	16310.749	16316.9338	5.232458479
	DE/best/2	16309.574	16295.889	16300.9111	4.21486466
Thyroid	DE/current-to-rand/1	16315.1	16300.865	16306.5729	4.953413693
	DE/current-to-best/1	16299.441	16294.082	16296.246	1.755794977
	DE/rand/1	1882.7504	1866.4769	1869.45039	4.93738597
	DE/rand/2	1933.1353	1904.8689	1922.52497	8.792795624
	DE/best/1	1892.2047	1879.5593	1883.38591	3.564562431
Breast Cancer	DE/best/2	1876.3129	1866.6946	1871.08952	2.978686652
	DE/current-to-rand/1	1896.8396	1877.1492	1888.41288	5.396856437
	DE/current-to-best/1	1884.3173	1867.5745	1874.91641	4.83995231
	DE/rand/1	2971.8862	2964.4321	2966.53782	2.650869494
	DE/rand/2	3311.5127	2965.7437	3012.85072	105.5592452
Pima	DE/best/1	3033.665	3000.5054	3018.80216	11.39918918
	DE/best/2	2977.0986	2966.4019	2971.18367	4.13411904
	DE/current-to-rand/1	2999.9495	2973.002	2987.84142	8.637445694
	DE/current-to-best/1	2974.674	2964.858	2969.47028	3.284382688
	DE/rand/1	47569.33	47561.23	47563.2065	3.092244609
Glass	DE/rand/2	47678.477	47564.79	47601.2339	38.04236253
	DE/best/1	47975.336	47780.234	47884.8688	54.80166752
	DE/best/2	47591.04	47562.61	47575.7867	10.42867773
	DE/current-to-rand/1	47614.9	47563.367	47587.5049	17.31487431
	DE/current-to-best/1	47572.508	47563.207	47566.7832	2.977584547
Ecoli	DE/rand/1	238.61061	214.88739	224.339602	7.936183946
	DE/rand/2	256.1896	239.43997	246.430862	4.351225392
	DE/best/1	233.55023	226.03458	229.798054	2.584396108
	DE/best/2	223.81511	217.97005	221.988005	1.830325338
	DE/current-to-rand/1	246.86244	241.1548	244.030698	2.155017781
Ecoli	DE/current-to-best/1	230.9455	218.51709	222.542509	4.262207958
	DE/rand/1	70.950424	64.20288	67.583721	2.17513627
	DE/rand/2	74.87054	71.46696	73.1026902	1.238720257
	DE/best/1	69.42949	66.73027	67.9053545	0.904321255
	DE/best/2	66.2538	64.51221	65.3631044	0.628261569
Ecoli	DE/current-to-rand/1	73.64982	71.09162	72.1286431	0.66660349
	DE/current-to-best/1	69.25693	65.226425	66.9274105	1.263528514

- Quantization error [16]: It calculates the average distance between data points and the cluster center as follows:

$$J_e = \frac{\sum_j^k [\sum_{d \in C_j} Dist(d, c_j) / |C_j|]}{k} \tag{11}$$

Where k indicate the number of clusters,  $C_j$  stands for the jth cluster, d is a data point in  $C_j$ ,  $c_j$  is the center of  $C_j$ ,  $|C_j|$  is the number of data points in  $C_j$  and  $Dist$  is the Euclidean distance between data point d and the center  $c_j$  of the cluster  $C_j$ . Lower quantization means the better cluster results.

#### 4.2 Experimental Results

The experimental results obtained by DE based clustering algorithms with different mutation strategies are summarized in Table 2-4. The qualities of solutions obtained by each algorithm are tabulated in terms of the worst, best, mean and standard deviation (Std.).

The vales of objective function obtained for all datasets are presented in Table 2. According to the mean values of the given results in Table 2, DE/rand/1 got better results for four datasets while DE/best/2 got better solutions for Iris and other two datasets with high number of clusters (Glass and Ecoli). The values of standard deviation obtained by DE/best/2 are smaller than these values got by other variants for Iris, Thyroid,

Glass, and Ecoli datasets. For Wine, Breast Cancer, and Pima datasets, the values of standard deviation achieved by DE/current-to-best/1 are smaller than the values acquired by others. It can be said that DE/best/2 and DE/current-to-best/1 is more robust than others where DE/current-to-best/1 achieved more stable results for high-dimensional datasets (in terms of number of feature and number of data instances) than DE/best/2.

Table 3. Comparison of sum of squared error

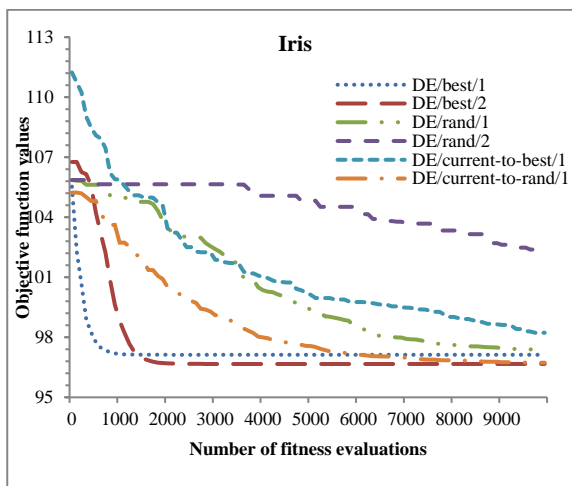
Dataset	Mutation Strategy	Worst	Best	Mean	Std
Iris	DE/rand/1	83.629715	80.42072	82.0800965	0.94004685
	DE/rand/2	89.5296	84.717865	87.5544574	1.842881411
	DE/best/1	81.711655	80.14506	80.9866801	0.654303667
	DE/best/2	80.38089	79.99243	80.1618585	0.113351464
	DE/current-to-rand/1	85.98539	81.2372	83.3685664	1.488298053
Wine	DE/current-to-best/1	80.505196	80.0851	80.3149833	0.154577131
	DE/rand/1	2594709	2513174.5	2564749	27068.79947
	DE/rand/2	2655556	2447204.2	2537712.62	70151.07796
	DE/best/1	2611303	2506935	2549237.9	31776.9966
	DE/best/2	2595641.8	2510130.2	2552362.65	27419.28448
Thyroid	DE/current-to-rand/1	2624745.8	2515277.8	2580781.38	32488.16905
	DE/current-to-best/1	2599812.5	2542793.5	2581293.87	21449.25797
	DE/rand/1	35247.93	34424.89	34884.4144	239.2168449
	DE/rand/2	36917.297	32058.793	35135.2568	1563.933731
	DE/best/1	36179.496	34523.406	35391.341	533.1252703
Breast Cancer	DE/best/2	35384.438	34568.34	35073.3447	272.6169406
	DE/current-to-rand/1	36213.64	34646.14	35288.382	537.5361905
	DE/current-to-best/1	35627.25	34905.703	35201.0054	237.0561105
	DE/rand/1	19609.496	19444.648	19503.1858	58.33649301
	DE/rand/2	22647.238	19426.004	19922.5529	973.8623906
Pima	DE/best/1	20586.848	19734.09	20163.7867	271.2584436
	DE/best/2	19747.975	19435.182	19558.3365	103.611039
	DE/current-to-rand/1	20101.18	19639.258	19871.8596	169.3518486
	DE/current-to-best/1	19715.303	19453.16	19568.6758	84.81113703
	DE/rand/1	5878486.5	5873893.5	5876531.25	1418.108702
Glass	DE/rand/2	5942486	5849599.5	5895572.6	25906.31329
	DE/best/1	6024383	5822446	5909666.1	57765.50274
	DE/best/2	5901308.5	5873936.5	5882881.75	8154.457415
	DE/current-to-rand/1	5966982	5869100.5	5916646.85	28769.17591
	DE/current-to-best/1	5882846	5872621.5	5878205.2	2936.693797
Ecoli	DE/rand/1	517.1698	352.20148	451.584795	51.7028711
	DE/rand/2	620.7019	490.95703	554.339182	39.59907819
	DE/best/1	534.244	470.30612	504.460414	21.25916904
	DE/best/2	511.1626	449.67932	479.885252	18.79054734
	DE/current-to-rand/1	588.00256	462.1183	521.978414	34.68964865
Ecoli	DE/current-to-best/1	506.72964	400.1598	467.785169	32.75475979
	DE/rand/1	18.250305	15.541359	17.0615069	1.064179566
	DE/rand/2	21.400398	19.9141	20.3151097	0.483599322
	DE/best/1	18.451756	15.876751	17.5507899	0.736882932
	DE/best/2	16.891884	15.218651	16.3252521	0.569443113
Ecoli	DE/current-to-rand/1	20.323498	18.774597	19.4565839	0.552628073
	DE/current-to-best/1	18.026596	16.368816	17.2151671	0.597203844

Table 3 and Table 4 summarized the quality of cluster solutions acquired from different mutation strategies in terms of the sum of squared error (SSE) and quantization error, respectively. According to the mean values given in both Table 3 and Table 4, the solutions acquired by random mutation strategies (DE/rand/1 and DE/rand/2) are better than other strategies for almost all of the test datasets. However, the mutation strategies that involve the best vector (DE/best/1, DE/best/2 and DE/current-to-best/1) obtained more stable results than others according to the standard deviation values given in both tables.

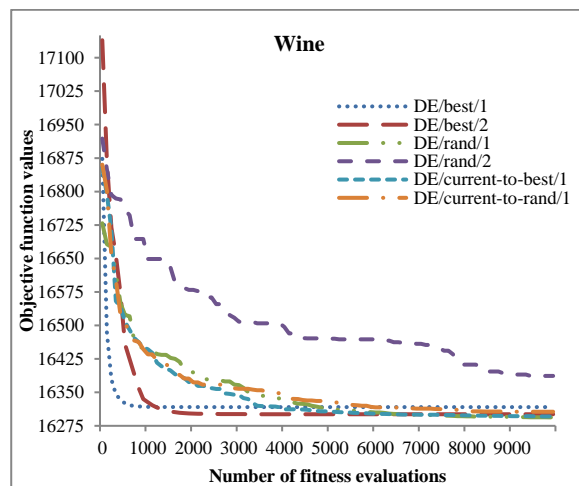
The convergence manners of different mutation strategies for all of the test datasets are shown in Figure 1. Based on the same 30 separate runs as mentioned above, the figure is illustrated with the averages of this runs. As observed in Figure 1, the convergence speed of DE/best/2 is the fastest on all datasets, whereas DE/rand/2 is the slowest and worst mutation strategy for all test datasets. Although DE/best/1 is faster than all variants except DE/best/2, it is not able to search for better solutions in the later stages, and it easily catches to local optima. DE/rand/1 finds better solutions for some datasets than other strategies in the late iterations, even though its convergence rate is slow in the early stages. DE/current-to-rand/1 can be regarded as a second-worst mutation strategy because it is slower and does not reach a better solution for all datasets except Iris. The exploration ability of DE/current-to-best/1 is not sufficient, and it does not catch a better solution for some datasets, although its convergence speed is a little fast.

Table 4. Comparison of quantization error

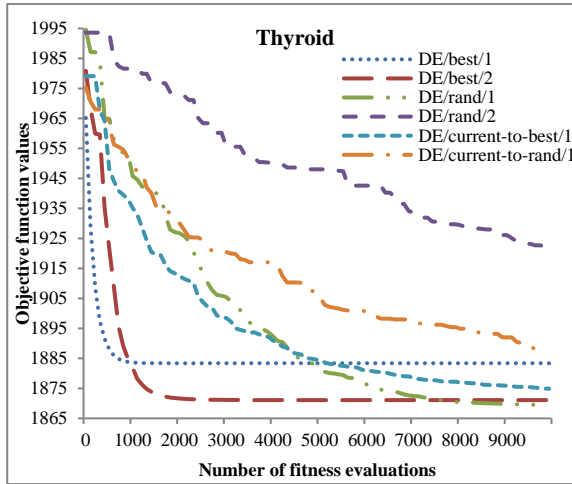
Dataset	Mutation Strategy	Worst	Best	Mean	Std
Iris	DE/rand/1	0.6516792	0.644194	0.648230088	0.002096196
	DE/rand/2	0.6917781	0.6613507	0.678133994	0.009085603
	DE/best/1	0.6529307	0.6434465	0.647316521	0.003001606
	DE/best/2	0.64522	0.6437538	0.643967262	0.00044599
	DE/current-to-rand/1	0.6572807	0.6491747	0.654163175	0.002575488
Wine	DE/current-to-best/1	0.6464462	0.6439006	0.644522413	0.00101929
	DE/rand/1	96.143974	95.56788	95.7610459	0.185679641
	DE/rand/2	97.008766	96.04605	96.4994756	0.295641799
	DE/best/1	96.29065	95.70694	96.0187704	0.211108131
	DE/best/2	96.216225	95.6381	95.8906245	0.181166983
Thyroid	DE/current-to-rand/1	96.16913	95.63155	95.8116934	0.182229921
	DE/current-to-best/1	95.86396	95.57987	95.668115	0.12394908
	DE/rand/1	9.302322	8.971374	9.09637795	0.115034208
	DE/rand/2	11.713471	9.0082445	9.77102625	0.171026345
	DE/best/1	9.33271	8.854203	9.15417135	0.161775128
Breast Cancer	DE/best/2	9.221522	9.0455675	9.14978185	0.064162164
	DE/current-to-rand/1	9.469611	9.058529	9.2730773	0.107299489
	DE/current-to-best/1	9.312383	8.98903	9.1844789	0.083235219
	DE/rand/1	5.21077	5.1888194	5.19391174	0.006918966
	DE/rand/2	5.765476	5.191242	5.272178	0.174804199
Pima	DE/best/1	5.351944	5.2500167	5.29663055	0.031849959
	DE/best/2	5.2154016	5.192358	5.20259006	0.00856287
	DE/current-to-rand/1	5.2407265	5.20192	5.22435062	0.012082721
	DE/current-to-best/1	5.2095737	5.189504	5.19876374	0.007111739
	DE/rand/1	67.81502	67.800385	67.803803	0.005570987
Glass	DE/rand/2	67.93898	67.770996	67.8345055	0.051236144
	DE/best/1	68.52382	67.7503	68.247306	0.221579355
	DE/best/2	67.8479	67.802315	67.822286	0.01679569
	DE/current-to-rand/1	67.82375	67.614174	67.7693949	0.067872086
	DE/current-to-best/1	67.81558	67.802895	67.8087031	0.004526859
Ecoli	DE/rand/1	1.7753247	1.1713182	1.42850018	0.216891528
	DE/rand/2	1.7360297	1.1389076	1.459802	0.189753391
	DE/best/1	1.6024362	1.2440042	1.36619334	0.120531122
	DE/best/2	1.5341662	1.1582086	1.32218297	0.097018546
	DE/current-to-rand/1	1.8478018	1.2105691	1.5327666	0.193121642
Ecoli	DE/current-to-best/1	1.6957501	1.2079638	1.38235998	0.156474922
	DE/rand/1	0.2327994	0.1982598	0.212012397	0.010604917
	DE/rand/2	0.2355672	0.2026296	0.221384451	0.008893089
	DE/best/1	0.2153705	0.1989665	0.205757483	0.004651351
	DE/best/2	0.2163881	0.1932458	0.203574999	0.008118651
Ecoli	DE/current-to-rand/1	0.2319987	0.199595	0.2186906	0.008642931
	DE/current-to-best/1	0.2503893	0.1883661	0.206791102	0.020067601



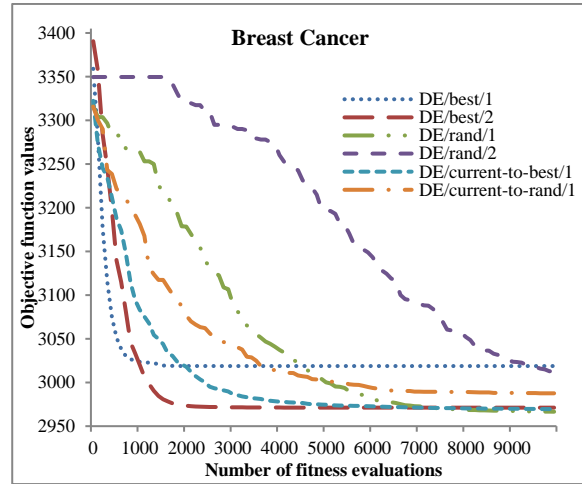
(a)



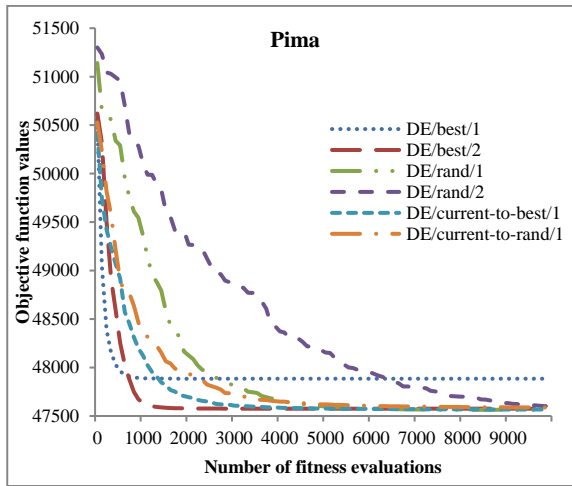
(b)



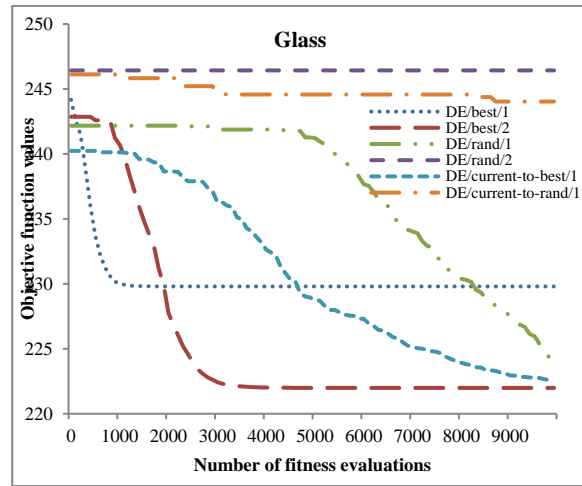
(c)



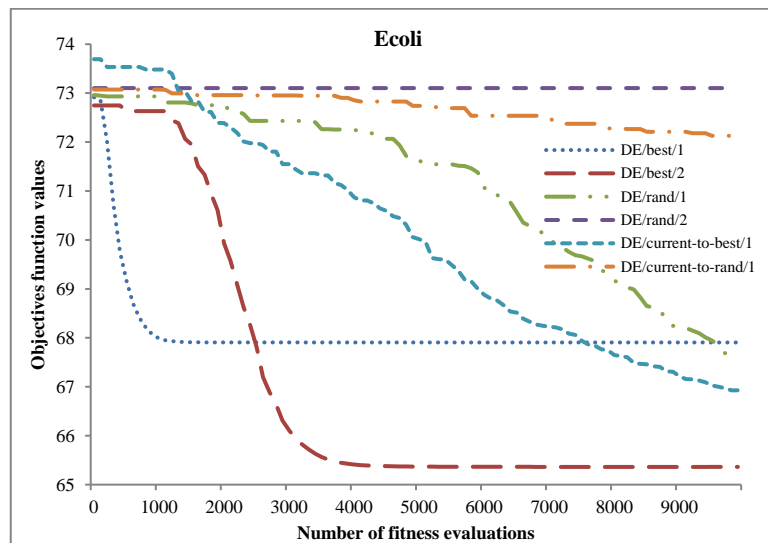
(d)



(e)



(f)



(g)

Figure 1. The convergence performance of different mutation strategies on six datasets



According to the overall experimental results, it is noticed that as follows: the mutation strategies based on the best solution are more robust and faster than others because these use the guidance information of the best solution to increase the exploitation ability and convergence speed of DE. Among them, DE/best/2 is more effective and robust for datasets with high number of clusters (Glass and Ecoli) due to the guidance information from the best solution and the application of two differentials. DE/rand/1 is able to find better solutions not only for high dimensional datasets (Breast Cancer and Pima) but also for moderate size of datasets because it can keep good diversity.

## 5. CONCLUSION

This paper presents an experimental investigation to analyze different mutation strategies of the DE algorithm for clustering problems. The performance of six mutation strategies has been tested on some UCI standard datasets mostly used in EAs based clustering. The quality of solutions and the convergence speed of different DE variants were compared to investigate the outcomes of the experiments. The experimental analysis pointed out that DE/rand/1 accomplishes to find better solutions for the moderate size of datasets. Besides, it shows good exploitation behavior in the later stages, while DE/best/2 shows good exploration behavior in the early stages. The test also showed that the mutation strategies that used the best solution achieve to find more stable results. Future work is to propose an effective mutation strategy for addressing large-scale clustering problems by applying the information from this experimental study.

## REFERENCES

- [1] A. P. Engelbrecht, *Computational Intelligence-An Introduction*, 2nd ed. England, John Wiley & Sons, 2007.
- [2] S. Das, P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no.1, pp. 4–3, 2010.
- [3] R. Storn, and K. Price. "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces." *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [4] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2008.
- [5] S. Paterlini, T. Krink, "High performance clustering with differential evolution," *Congress on Evolutionary Computation (CEC2004)*, pp. 2004–2011, 2004.
- [6] S. Paterlini, T. Krink, "Differential evolution and particle swarm optimisation in partitional clustering," *Computational Statistics & Data Analysis*, vol. 50, no. 5, pp. 1220–1247, 2006.
- [7] W.L. Xiang, N. Zhu, S.F. Ma, , X.L. Meng, and M.Q. An, "A dynamic shuffled differential evolution algorithm for data clustering," *Neurocomputing*, vol. 158, pp.144-154, 2005.
- [8] L. Zhang, L. Mao, H. Gong and H. Yang, "A K-harmonic means clustering algorithm based on enhanced differential evolution," *2013 Fifth International Conference on Measuring Technology and Mechanics Automation*, pp. 13–16, 2013.
- [9] W. Kwedlo, "A clustering method combining differential evolution with the K-means algorithm," *Pattern Recognition Letters*, vol. 32, no. 12, pp. 1613–1621, 2011.
- [10] L. Mao, H. J.Gong, and X. Y. Liu, "A K-means Clustering Algorithm Based on Enhanced Differential Evolution," in *Advanced Materials Research*, vol. 339, pp. 71–75, 2011.
- [11] E.R. Hruschka, R.J. Campello, and A.A. Freitas, "A survey of evolutionary algorithms for clustering," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 39, no. 2, pp. 133–155, 2009.
- [12] A.S. Shirkorshidi, S. Aghabozorgi, and T.Y. Wah, "A comparison study on similarity and dissimilarity measures in clustering continuous data," *PloS one*, vol. 10, no. 12, 2015.
- [13] <https://archive.ics.uci.edu/>
- [14] S.J. Nanda, and G. Panda, "A survey on nature inspired metaheuristic algorithms for partitional clustering," *Swarm and Evolutionary computation*, vol. 16, pp. 1-18, 2014.
- [15] S. Rahnamayan, H.R. Tizhoosh, and M.M.A. Salama, "Opposition-based differential evolution," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 64–79, 2008.
- [16] H.T.T. Thein, and K.M.M. Tun, "Evaluation of differential evolution and K-means algorithms on medical diagnosis," in *5th National Symposium on Information Technology: Towards New Smart World (NSITNSW)*, pp. 1-4, 2015.

**BIOGRAPHY OF AUTHORS**

**Pyae Pyae Win Cho** received the M.C.Sc degree from Computer University, Patheingyi, Myanmar in 2012. She is an Assistant Lecturer at the University of Computer Studies, Myeik, Myanmar. She is currently working toward a Ph.D. degree at the University of Computer Studies, Yangon, Myanmar. Her research interests include evolutionary computing and data mining.



**Thi Thi Soe Nyunt** got B.Sc. Physics (Hons:) degree from Yangon University in 1994 and got Master of Information Science (M.I.Sc.) degree and Ph. D (IT) from UCSY in 1998 and 2004 respectively. She is currently working as a Professor and Head of department in Faculty of Computer Science, UCSY. Her research interests include Knowledge & Software Engineering, Database, Computer Graphics, Big Data Analytics, Artificial Intelligence and Neural Network.