An Improved Differential Evolution Algorithm with Opposition-Based Learning for Clustering Problems

Pyae Pyae Win Cho
University of Computer Studies, Yangon
pyaepyaewincho@ucsy.edu.mm

Thi Thi Soe Nyunt
University of Computer Studies, Yangon
thithi@ucsy.edu.mm

Abstract

Differential Evolution (DE) is a popular efficient population-based stochastic optimization technique for solving real-world optimization problems in various domains. In knowledge discovery and data mining, optimization-based pattern recognition has become an important field, and optimization approaches have been exploited to enhance the efficiency and accuracy of classification, clustering and association rule mining. Like other population-based approaches, the performance of DE relies on the positions of initial population which may lead to the situation of stagnation and premature convergence. This paper describes a differential evolution algorithm for solving clustering problems, in which opposition-based learning (OBL) is utilized to create high-quality solutions for initial population, and enhance the performance of clustering. The experimental test has been carried out on some UCI standard datasets that are mostly used for optimization-based clustering. According to the results, the proposed algorithm is more efficient and robust than classical DE based clustering.

Keywords: *differential evolution algorithm, clustering, opposition-based learning*

I. INTRODUCTION

The rapid progress in technologies for data storage and the remarkable increase in internet applications have made a huge amount of different types of data. Undoubtedly, these data involve a lot of useful information and knowledge. Data mining is an efficient way to extract valuable hidden patterns from these large data sets. Clustering, often called unsupervised learning, is one technique for finding intrinsic structures from data with no class labels. It partitions a given dataset into different sets called clusters such that members of a cluster are more similar to each other and these are dissimilar from members in other clusters. Cluster analysis has been successfully used in various domains such as image processing, web mining, market segmentation, medical diagnosis, etc. The various kinds of clustering approaches have been proposed and used in different research communities [1]. Partition-based and hierarchy-based clustering are the two most important approaches [2]. In partition-based clustering, the data instances are divided into a given number of different partitions based on their similarities. In hierarchy-based clustering, a nested sequence of partitions is produced by representing as a dendrogram. This paper emphasizes on the partition-based clustering. The most well-known and widely used partition-based clustering approach is K-means [3]. Nevertheless, it is sensitive to initial settings and may trap to local optima. In recent times, optimization-based clustering approaches have become an attractive way in solving cluster analysis problems [4] [5] due to their population-based, self-organized and decentralized search behavior and their ability of discovering superior results.

DE is a simple, efficient and robust optimizer for many real-world global optimization problems in various domains. It has been successfully utilized as an effective alternative way to solve clustering problems [6] [7]. DE, however, sometimes fails to meet the global optimum. It may suffer from the situation of stagnation in which DE may stop searching the global optimal solution even though it has not caught the local optima. DE is vulnerable to premature convergence which may take place due to the loss of diversity in population. Besides, DE's performance relies on control parameter settings and the positions of initial population. If the initial population is composed of high-quality individuals, it is more likely to give rise to higher quality or acceptable solution. Many research works have been recently introduced to boost the efficiency of standard differential evolution algorithm [8] [9]. This paper aims to employ the DE algorithm for clustering problems. In order to boost the clustering effectiveness of DE based algorithm, a two-step population

initialization method (OBL) is utilized to create the high-quality initial population.

The rest sections are structured as follows: In section 2, a background about the basic concepts related to the canonical DE algorithm and OBL approach, and existing works related to cluster analysis utilizing DE are presented. The proposed approach is introduced in Section 3 and the conducted experimentations for evaluating the clustering performance of the DE variants are presented in Section 4. Lastly, Section 5 describes the conclusion of this paper.

II. BACKGROUND

In this section, the brief description of DE algorithm and OBL scheme, and then the related works to cluster analysis based on DE are described.

A. Differential Evolution Algorithm

Differential evolution (DE) algorithm, proposed by R. Storn and K. Price in 1995, is a simple and dominant population-based nature-inspired approach to solve global optimization problems [10]. Several variations of the DE algorithm have been recently introduced and employed to resolve optimization problems in several domains. DE, like other population-based algorithms, evolves the population of solutions at each generation by reproduction processes to deliver a better solution. The procedure of standard DE algorithm involves four successive steps such as initialization of a population, mutation, crossover, and selection. As soon as the first step is performed by generating an initial population, DE performs three remaining steps iteratively until a stopping situation. A brief description of all these steps based on the traditional DE [11] is presented in the followings.

1) Population Initialization

DE generally constructs the initial population with a set of candidate solution vectors (also called as chromosomes or individuals) that are randomly selected from the search space. Each solution vector X_i in the population, $P=\{X_1, X_2, .., X_{NP}\}$ at the t^{th} iteration is denoted as $X_{i,g}=\{x_{i,t}^1, x_{i,t}^2, \ldots, x_{i,t}^d\}$ where NP represents the number of population and d refers to the number of solution dimensions.

2) Mutation

Once the population is initialized, a mutant vector V_i is created for each parent vector X_i by perturbing a target vector with a weighted difference of two random solution vectors from the current population as stated by the following equation:

$$V_i=X_a+f(X_b-X_c) \quad (1)$$

where X_a, $X_b$and X_c are three randomly selected vectors such that a, b, c $\in$ [1, NP] and i $\neq$ a $\neq$ b $\neq$ c, and then f is the scaled factor within (0, ∞).

3) Crossover

In crossover step, an offspring vector U_i is created by recombining the parent X_i and the mutant V_i. Crossover is implemented as follows:

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

where rand(j) $\in$ U(0,1), i$\in$[1,NP], j$\in$[1,d], and CR is the crossover rate within (0,1).

4) Selection

The selection phase determines the survival solution among the parent and offspring vectors according to the value of fitness function. For a maximization problem, the solution with the larger value of fitness will survive in the iteration as follows:

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

where f(.) indicates the value of fitness function.

There are numerous variants that were extended from basic DE. These variants are denoted by DE/x/y/z notation where x indicates the way of choosing a target vector; y specifies the number of pairs of vectors to compute difference vectors, and the last symbol, z indicates the recombination scheme for the crossover operator [11]. Normally, the DE algorithm's performance highly relies on control parameters, adopted mutation strategy, and population size and positions.

B. Opposition-Based Learning

An innovative scheme for machine intelligence algorithms, Opposition-based learning (OBL) has been proposed by Tizhoosh and utilized to accelerate genetic algorithm (GA), artificial neural networks (ANN), reinforcement learning [12] [13], and differential evolution algorithm [14] [15]. Numerous researches have carried out the integration of population-based optimization approaches with the OBL scheme to enhance their search behaviors. The key idea of OBL is searching for a superior estimation of the current candidate solution by considering estimates and their

respective opposite estimates together. Definition 1 and 2 are the concept of OBL described in [12].

Definition 1- If x be a real number in a range of $[x_{min}, x_{max}]$, then the opposite number of x, $\breve{x}$ can be defined as follows:

$$\breve{x}=x_{min}+x_{max}-x \quad (4)$$

Definition 2- If $P(x_1, x_2,\ldots, x_n)$ is a point in n-dimensional space such that $(x_1, x_2,\ldots, x_n)\in R$ and $x_i\in[x_{i,min}, x_{i,max}]$, then the opposite point of P, $\breve{P}(\breve{x_1}, \breve{x_2},\ldots,\breve{x_n})$ can be defined by its elements as follows:

$$\breve{x_i}=x_{i,min}+x_{i,max}-x_i \quad (5)$$

Opposition-based optimization generates an opposite solution for each candidate solution. After calculating their fitness values, the fitter one among the candidate and its opposite will survive in the evolution process. This paper uses this concept to find initial cluster solution for DE.

C. Related Works

The application of differential evolution to the cluster analysis has been an interesting research topic for a long time. The work proposed by S. Paterlini and T. Krimk [6] can be regarded as innovative effort in this field. The authors investigated the clustering performance of DE, GA, and particle swarm optimization (PSO) by employing medoid-based representation. Finally, they concluded that DE is superior compared to the other approaches and more suitable rather than GA for clustering analysis. In [16], the authors proposed a dynamic shuffled differential evolution algorithm for data clustering (DSDE) to enhance the speed of clustering convergence. This work proposed a random multi-step sampling as an initialization method due to most of clustering algorithms are sensitive to the chosen initial centroids, which can lead to premature convergence. They also applied a sorting and shuffled scheme to divide the whole population into two subpopulations in order to enhance the diversity of population. In this proposed approach, DE/best/1 mutation strategy is employed for both subpopulations during the process of evolution to exchange the direction information among the populations effectively and to balance the exploitation ability of this mutation strategy. According to the results, DSDE is superior to the classical DE and other well-known evolutionary algorithms in term of total intra-cluster distances.

The differential evolution based clustering approach with K-mean algorithm (DE-KM) [17] was proposed to catch high quality clustering solutions in term of sum of squared errors (SSE). In this work, the K-mean algorithm was incorporated into the process of DE to create the initial population and optimize the offspring solution. Their reported experimental results described that the incorporation of DE with a local search algorithm is superior to the DE only. In [18], an efficient data clustering approach based on DE was presented to manage the weaknesses of k-means algorithm. The proposed work utilized the classical DE with the within-cluster and between-cluster distances as objective functions. They concluded that their presented approach was comparable to the K-means and achieved better solutions. In [19], the author proposed a differential evolution algorithm with macromutations (DEMM) to enhance the exploration ability of the classical DE for clustering. In DEMM, the macromuation scheme was applied with the application probability and macromutation intensity that are dynamically changed during the evolution process. The application probability was used to shift between the common mutation and crossover and macromutaions. The intensity (crossover rate) of the macromutations was exponentially decreased in order to get wider exploration at the initial stage and then gradually turn into better exploitation at the later stage. As the performance of the DE algorithm depends on the adopted mutation strategies, a new DE variant, Forced Strategy Differential Evolution (FSDE) was proposed and applied it for data clustering in [20]. In FSDE, a new mutation strategy was presented, which applied two difference vectors based on the best solution vector. Besides a constant traditional scaling factor, this strategy used an additional variable control parameter. FSDE applied the result from K-means as one member of the initial population and then chose the rest of the population randomly. According to the stated results, the FSED delivered fine cluster solutions based on different cluster validity measures.

The main intention of this paper is to boost the clustering performance of the DE algorithm by adapting a two-steps initialization method. Opposition-based learning proposed to accelerate the machine intelligence algorithms will be used as an initialization method.

III. A Differential Evolution based Clustering Algorithm with Opposition-Based Learning

For applying the DE algorithm to the clustering problem, a chromosome encodes a cluster solution that is represented by vectors of real numbers. Hence, the length of each chromosome depends on the dimension of the given dataset and the number of clusters in the dataset. If K is the number of clusters and d is the number of dimensions of the dataset, K*d will be the length of the chromosome , where the first d-genes of the chromosome represents the first cluster solution, the next d- genes encodes the second solution, and the last d-genes is the K^{th} cluster solution as shown in Fig.1.

In the optimization based clustering problem, cluster validity measures are used as objective functions. In this paper the fitness of every chromosome is calculated by using the total intra-cluster distance (IntraD) which is formulated as follows:

$$\text{IntraD} = \sum_{j=1}^{k} \sum_{p \in C_j} d(p, c_j) \qquad (6)$$

where k is the number of clusters, p is a data point in the j^{th} cluster C_j, c_j is the cluster center of C_j and then $d(p, c_j)$ denotes the Euclidean distance between data point and cluster center of C_j. In this work, a two-step initialization technique is adopted in order to get better positions of the initial population. The opposite-based learning is exploited to enhance the quality and diversity of initial population. To generate the initial population, each solution vector $X_i=\{x_i^1,\ldots, x_i^d, \ldots, x_i^{(k-1)d+1}, \ldots, x_i^{kd}\}$ is firstly initialized with randomly selected k data point from the given dataset. And then its opposite vector $\breve{X}_i=\{\breve{x}_i^1, \ldots, \breve{x}_i^d, \ldots, \breve{x}_i^{(k-1)d+1}, \ldots, \breve{x}_i^{kd}\}$

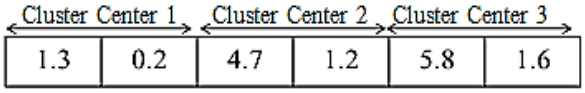


Figure 1. Chromosome Encoding for a Cluster Solution

is calculated according to the equation (5). The fitness of each solution vector and its opposite vector are evaluated and then, the fitter vector is selected as an initial solution. Once the initial population is constructed, the evolution processes are accomplished until a stopping condition is met. In Fig. 2, the proposed DE based clustering algorithm, DEC-OBL is given.

Algorithm: DEC-OBL
Input: Dataset (D), Number of clusters (k), Maximum iteration (maxIt), Number of population (NP), Scaling factor (F), Crossover rate (Cr)
Output: Cluster centers
//Population Initialization with OBL scheme
For i=1 to NP
1. Initialize a chromosome with k randomly selected data points
2. Calculate the opposite chromosome according to eq. (5)
3. Evaluate the fitness of two chromosomes according to eq. (6)
4. Select the fitter one from the pair of chromosomes

End
// Evolution Process
For i=1 to maxIt
1. Generate the mutant vector by applying the mutation operation
2. Generate the offspring vector by applying the binomial crossover operator
3. Evaluate the fitness offspring vector
4. Update the population by selection operation

End

Figure 2. Differential Evolution based Clustering Algorithm with Opposition-Based Learning

IV. Experimentation

This section provides the computational results of the proposed approach. The aim is to study the impact of population initialization technique on the clustering performance of the DE based approaches. The proposed algorithm and the traditional DE based clustering algorithm (with random initialization method) were evaluated on two mutation strategies, DE/best/1and DE/rand/1. These algorithms were implemented on Core i7 processor, 8GB RAM, and 64-bit operating system using the java programming language (NetBean IDE 8.2).

The experimental test was conducted on a number of the mostly used UCI benchmark data sets for optimization-based clustering [5]. The summary of the used datasets is presented in Table I. The control parameters were set as follows: the size of the population = 100, the number of maximum iterations = 100, the scaling factor, F = 0.9, and the crossover rate, Cr = 0.5. Each approach was independently run 30 times on each dataset.

In Table II, the obtained results are given in terms of maximum, minimum, mean and standard deviation. As reported in Table II, DEC-OBL is superior to DE on both mutation strategies. The

adaptation of the OBL based initialization method can enhance the quality of cluster solutions. Moreover, DEC-OBL achieved the smaller values of the standard deviation on all data sets. Thus, the proposed approach is more effective and robust than DE.

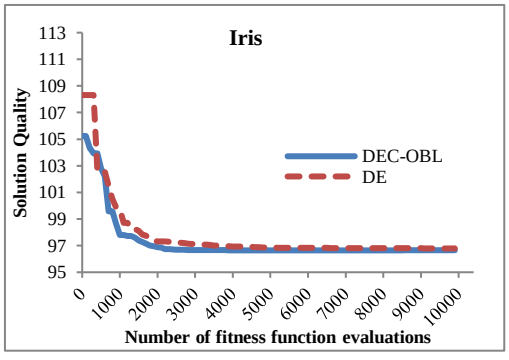
The convergence rate of DE and DEC-OBL with the DE/best/1 mutation strategy is shown in Fig. 3. As may be noticed from figure 3, the convergence rate of DEC-OBL is slightly faster than DE, and DEC-OBL achieves a better exploration of the search space at the early stage of the searching.

TABLE I. THE DESCRIPTION OF DATASETS

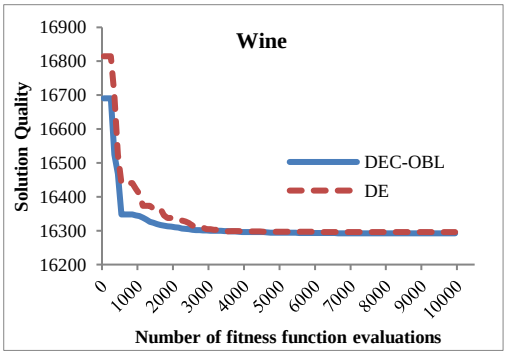
Datasets	No. of features	No. of data instances	No. of Clusters
Iris	4	150	3
Wine	13	178	3
Glass	9	214	6
Cancer	9	683	2
Thyroid	5	215	3

TABLE II. COMPARISON FOR SUM OF INTRA-CLUSTER DISTANCE OF DE AND DEC-OBL WITH TWO MUTATION STRATEGIES

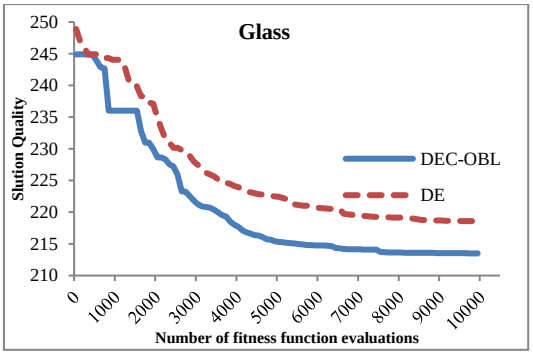
Datasets	Mutation Strategies	Algorithms	Maximum	Minimum	Mean	Std.
Iris	DE/rand/1	DE	102.1521	101.475162	99.4258	0.794783153
		DEC-OBL	100.2946	97.875	99.70693	0.018106804
	DE/best/1	DE	97.4226	96.84346	96.6578	0.273315338
		DEC-OBL	96.6578	96.65593	96.6554	0.00071655
Wine	DE/rand/1	DE	16342.188	16311.866	16319.52917	9.278565097
		DEC-OBL	16312.8125	16300.004	16306.56595	4.709555761
	DE/best/1	DE	16313.771	16295.3475	16298.85414	5.527263155
		DEC-OBL	16295.817	16292.639	16294.61315	1.221286072
Glass	DE/rand/1	DE	248.6747	239.069	243.760621	3.651353923
		DEC-OBL	242.1051	235.286	239.78097	2.258780807
	DE/best/1	DE	223.29367	218.3881	220.372695	1.823091664
		DEC-OBL	217.4788	213.3792	215.410262	1.196080611
Cancer	DE/rand/1	DE	3008.2185	2989.3984	2998.99096	8.07684039
		DEC-OBL	2988.9023	2977.8376	2983.9711	3.59454961
	DE/best/1	DE	2984.6045	2965.6616	2968.92243	5.64962512
		DEC-OBL	2965.309	2964.4873	2964.72553	0.30670541
Thyroid	DE/rand/1	DE	1898.324	1892.0066	1895.27901	2.39391237
		DEC-OBL	1882.9974	1878.0023	1881.18675	1.54906751
	DE/best/1	DE	1897.0674	1869.5518	1887.54791	9.17279202
		DEC-OBL	1869.4032	1866.5364	1867.16008	1.10661843



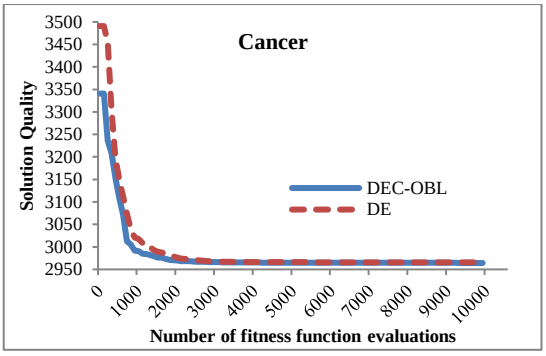
(a)



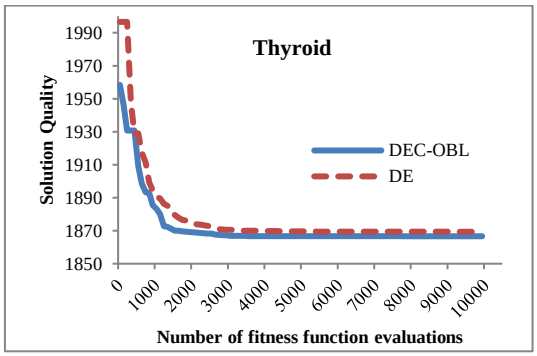
(b)



(c)



(d)



(e)

Figure 3. Convergence Performance of DE and DEC-OBL with DE/best/1 Mutation Strategy on Datasets: (a) Iris, (b) Wine, (c) Glass, (d) Cancer, (e) Thyroid

V. CONCLUSION

In this paper, a DE based clustering algorithm is presented. The idea of OBL has been exploited to enhance the clustering performance of DE. OBL is used for generating the initial solutions instead of selecting randomly. According to the obtained results, the proposed algorithm achieves the better cluster solutions and it is more robust than classical DE based clustering. As future works, the convergence speed of the proposed algorithm will be enhance by dynamically adjusting the scaling factor and crossover rate, and different cluster validity measures will be considered as fitness function. Moreover, the effect of population size on the cluster results will be investigated.

REFERENCES

[1] A. K. Jain, "Data clustering: 50 years beyond K-means", Pattern recognition letters, vol. 3(8), June 2010, pp. 651–666.

[2] B. Liu, Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, Springer Science & Business Media, 2007.

[3] J. MacQueen, "Some methods for classification and analysis of multivariate observations", Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967, vol. 1(14), pp. 281–297.

[4] 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, 2009, vol. 39(2), February 2009, pp. 133-155.

[5] S.J. Nanda, G. Panda, "A survey on nature inspired metaheuristic algorithms for partitional clustering", Swarm and Evolutionary Computation, vol. 16, June 2014, pp. 1-18.

[6] S. Paterlini, T. Krink, "High performance clustering with differential evolution", Proceedings of the 2004 Congress on Evolutionary Computation, June 2004.

[7] S. Paterlini, T. Krink, "Differential evolution and particle swarm optimisation in partitional clustering", Computational Statistics & Data Analysis, 2006, 50(5), pp. 1220–1247.

[8] S. Das, P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art", IEEE Transactions on Evolutionary Computation, vol. 15(1), October 2010, pp. 4-31.

[9] S. Das, S.S. Mullick, P.N. Suganthan, "Recent advances in differential evolution – An updated survey", Swarm and Evolutionary Computation, vol. 27, April 2016, pp. 1-30.

[10] R. Storn, K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces", Journal of Global Optimization, vol. 11(4), December 1997, pp. 341–359.

[11] A. P. Engelbrecht, Computational Intelligence-An Introduction, Second Edition, John Wiley & Sons Ltd, England, 2007.

[12] H.R.Tizhoosh. "Opposition-Based Learning: A New Scheme for Machine Intelligence", International Conference on Computational Intelligence for Modeling Control and

Automation -MCA'2005, Vienna, Austria, vol. I, 2005, pp. 695-701.

[13] H. R. Tizhoosh, "Opposition-Based Reinforcement Learning", Journal of Advanced Computational Intelligence and Intelligent Informatics, vol.10 (4), 2006, pp. 578-585.

[14] S. Rahnamayan, H. R. Tizhoosh, M. M. A. Salama, "Opposition-Based Differential Evolution", IEEE Transactions on Evolutionary Computation, vol. 12(1), February 2008, pp. 64-79.

[15] S. Rahnamayan, H.R. Tizhoosh, M.M.A. Salama,"Opposition-Based Differential Evolution (ODE) with Variable Jumping Rate," IEEE Symposium on Foundations of Computational Intelligence, April 2007.

[16] W.-l. Xiang, N. Zhu, S.-f. Ma, X.-l. Meng, M.-q. An, "A dynamic shuffled differential evolution algorithm for data clustering", Neurocomputing, vol. 158, June 2015, pp. 144-154.

[17] W. Kwedlo, "A clustering method combining differential evolution with the K-means algorithm", Pattern Recognition Letters, vol. 32 (12), September 2011, pp. 1613-1621.

[18] M. Hosseini, M. Sadeghzade, R. Nourmandi-Pour, "An efficient approach based on differential evolution algorithm for data clustering, Decision Science Letters, vol. 3 (3), June 2014, pp. 319-324.

[19] G. Martinović, D. Bajer, "Data Clustering with Differential Evolution Incorporating Macromutations", Swarm, Evolutionary, and Memetic Computing - 4th International Conference, SEMCCO 2013, Chennai, India, December 2013, Proceedings, Part I, Springer International Publishing, Cham, pp. 158-169.

[20] M. Ramadas, A. Abraham, S. Kumar, "FSDE-Forced Strategy Differential Evolution used for data clustering", Journal of King Saud University-Computer and Information Sciences, vol.31 (1), January 2019, pp. 52-61.