Received: June 5, 2020. Revised: July 27, 2020.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Data Clustering based on Modified Differential Evolution and Quasi-Opposition-based Learning

Pyae Pyae Win Cho[1]* **Thi Thi Soe Nyunt[1]**

[1] *University of Computer Studies, Yangon, Myanmar*
* Corresponding author's Email: pyaepyaewincho@ucsy.edu.mm

Abstract: Differential Evolution (DE) has become an advanced, robust, and proficient alternative technique for clustering on account of their population-based stochastic and heuristic search manners. Balancing better the exploitation and exploration power of the DE algorithm is important because this ability influences the performance of the algorithm. Besides, keeping superior solutions for the initial population raises the probability of finding better solutions and the rate of convergence. In this paper, an enhanced DE algorithm is introduced for clustering to offer better cluster solutions with faster convergence. The proposed algorithm performs a modified mutation strategy to improve the DE's search behavior and exploits Quasi-Opposition-based Learning (QBL) to choose fitter initial solutions. This mutation strategy that uses the best solution as a target solution and applies three differentials contributes to avoiding local optima trap and slow convergence. The QBL based initialization method also contributes to increasing the quality of the clustering results and convergence rate. The experimental analysis was conducted on seven real datasets from the UCI repository to evaluate the performance of the proposed clustering algorithm. The obtained results showed that the proposed algorithm achieves more compact clusters and stable solutions than the competing conventional DE variants. Moreover, the performance of the proposed algorithm was compared with the existing state of the art clustering techniques based on DE. The corresponding results also pointed out that the proposed algorithm is comparable to other DE based clustering approaches in terms of the value of the objective functions. Therefore, the proposed algorithm can be regarded as an efficient clustering tool.

Keywords: Differential evolution, Clustering, Mutation strategy, Quasi-opposition-based learning.

1. Introduction

There has been an enormous growth in the amount of data being generated from different sources in the era of information technology. It is needed to alter this raw data into useful information for various applications. Data mining, also known as Knowledge Discovery, is a manner of drawing out valuable knowledge and hidden patterns from raw data [1]. Data mining techniques have been majorly categorized into two types, such as supervised learning, which trains a model on labeled data and predicts the label of new data, and unsupervised learning, which explores hidden patterns and relationships in unlabeled data. Clustering is an unsupervised learning technique that partitions a dataset into meaningful sets called clusters based on the dissimilarity or similarity between data objects such that data objects in a cluster are more related than those in others. There are two types of clustering techniques, namely hard clustering and soft clustering. Hard clustering techniques find partitions of a dataset by inserting each data object into only one group, whereas soft clustering techniques can assign data objects into more than one cluster based on their different weights or likelihoods.

Clustering has been broadly adopted in several applications such as market research, image processing, pattern recognition, etc. [2, 3]. Several clustering algorithms have been proposed and employed in different domains. They are mainly classified into hierarchical and partitional clustering. The hierarchical clustering algorithm discovers clusters either in bottom-up fashion (known as agglomerative approach) or in top-down fashion (known as divisive approach). The agglomerative approach considers each data object as a distinct group and continuously combines the more similar pairs of groups into a larger group. The divisive approach considers the entire dataset as a cluster and divides it into smaller groups, recursively. Single-link and complete-link are two of the most famous hierarchical algorithms. Partitional clustering algorithms identify a predefined number of non-overlapping groups together. K-means is one of the most straightforward and famous partitional clustering algorithms [2-4].

Several nature-inspired metaheuristics have been recently introduced and widely used in numerous applications. Two major kinds of nature-inspired metaheuristics are swarm intelligence (SI) and evolutionary algorithms (EAs). The application of nature-inspired metaheuristics to clustering has become an attractive research topic in data mining. Researchers have recently designed various metaheuristics-based algorithms in this field [5-7]. Differential Evolution (DE) algorithm is one of the most prominent nature-inspired metaheuristic algorithms successfully used in clustering. DE is introduced by Storn and Price in 1995, which is a simple and efficient population-based optimization algorithm, and has been well and widely employed in many real-world problems [8]. The effectiveness and achievement of the DE algorithm are determined by the positions of the initial population, the adopted mutation and crossover strategies, and control parameters. Several DE variants are proposed and designed by modifying the population initialization, adopted strategies and setting of control parameters.

The aim of this paper is to propose a DE variation for hard partitional clustering that provides better cluster solutions with faster convergence. In this work, the modified mutation strategy is presented for harmonizing the exploitation and exploration ability of the DE algorithm. This strategy uses the best solution as a target vector to get more exploitation ability, one differential between the best and current vector to lead to the good convergence characteristics, and two differentials between three random vectors to increase the exploration power of DE. Besides, the population initialization technique based on Quasi-Opposition-based Learning (QBL) is also applied for improving the clustering performance of the proposed variant due to keeping fitter solutions for the initial population raises the probability of finding better solutions and the rate of convergence.

The structure of this paper is as follows: Section 2 gives the works related to the application of DE in clustering; Section 3 and 4 provides the basic idea of the standard DE algorithm and QBL scheme; Section 5 explains the proposed approach; Section 6 gives the carried out tests for the comparison of clustering performance; Section 7 finishes the paper with a conclusion.

2. Related work

The utilization of the DE algorithm in clustering has become an attractive research subject for a long time. In various real-world applications, different DE variants have been used to employ clustering independently or incorporate within the existing clustering approaches. An improved variant of the conventional DE algorithm for the automatic clustering problem was introduced in [9]. The authors improved the population initialization step of the DE algorithm by embedding with a cluster decomposition algorithm (CDA). Moreover, the proposed algorithm dynamically self-adjusted the scaling factor and the crossover rate. The authors also proposed four improvement scenarios to update chromosomes for evolving the population. The updating rules utilized the best solution effect and the acceleration mechanism for the faster convergence, and the handling downhill concept for leading to a better solution. Moreover, the saturated solution is also applied to maintaining population diversity. The work proposed in [10] is also for automatic clustering. In this work, an adaptive DE algorithm was combined with the neighborhood search (NS). The adaptive approach was operated to fine-tune the control parameter of DE, and NS was utilized to control the diversification of solutions. The proposed algorithm applied a new mutation approach in NS based on the success rate of DE and NS to balance the search ability. In [11], a dynamic shuffled differential evolution algorithm (DSDE) was offered for the improvement of the convergence performance of data clustering algorithms. In the proposed work, a novel random multistep sampling initialization method was incorporated to overcome the premature convergence, and a dynamic sorting and shuffled technique was also integrated to split the total population into two subpopulations for improving the population diversity. DE/best/1 mutation scheme was applied to both subpopulations that exchange the guidance information to adjust the exploitation and exploration ability of DSDE.

In [12], a new DE variant with a new mutation strategy, Forced Strategy Differential Evolution (FSDE) was presented, and the application of FSDE on data clustering was also presented. In the proposed new mutation strategy, a variable parameter, besides the traditional scaling factor, was applied to enhance the quality of the donor vector and hence, the effectiveness of the algorithm. For the implementation of FSDE in clustering, the outcome of the K-means algorithm was applied as an initial solution, and the remaining ones of the initial population were selected randomly. In [13], a variance-based differential evolution algorithm with an optional crossover for data clustering (VDEO) was proposed to adjust the search behaviors of the standard DE algorithm, and enhance its clustering efficiency. In the proposed algorithm, a single-based solution scheme was implemented instead of the population concept to reduce the computation cost for the objective function evaluations of the solutions in the population. VDEO employed DE/best/1 and DE/rand/1 mutation strategies with a switchable mechanism to control the search processes, and dynamically estimated the mutation factor by using a proposed vector-based technique. Besides, an optimal crossover strategy was presented and employed, in which the objective function value (fitness) of the mutant vector is also taken into account to generate the offspring vector, and the crossover rate is dynamically adjusted. The performance comparison of DE with local search algorithms for clustering problems was presented in [14]. The authors compared the effectiveness of DE with chaotic local search (CLS), levy flight (LF), and golden section search (GSS) in terms of solution quality and convergence speed. They concluded that DE-LF is simple and potential for hard partitional clustering problems.

An adaptive unified differential evolution (AuDE) was applied for optimal clustering in [15]. AuDE utilized a unified mutation strategy that is the combination of the two most used standard mutation strategies and also used an adaptive approach to adjust the scale factor and crossover rate. Another algorithm combining the DE algorithm and K-means for optimal clustering was presented in [16]. This proposed algorithm also applied K-means on the solution vectors created from the DE recombination processes. A heuristic reordering procedure of the cluster centers was introduced to improve the process of classification. In [17], a differential evolution algorithm with macromutations (DEMM) was proposed for data clustering. In DEMM, macromutations were applied instead of the traditional DE recombination processes. There was a set probability (the application probability and macromutation intensity) in the proposed macromutations. The linearly increased application probability managed to switch between the standard reproduction processes and the macromutations, and the exponentially decreased intensity (crossover rate) controlled the macromutations to generate the offspring. This dynamically adjusted set probability provided a good symmetry within the exploitation and exploration processes of DE.

The problems to be taken into account and addressed to develop an efficient and robust clustering algorithm are as follows. The positions of initial solutions impact the performance of the DE algorithm. The fitter initial solutions lead to reach a better solution faster. Moreover, the adopted mutation strategy is critical for the search ability of the algorithm to overcome local optimal trap and slow convergence. The various approach proposed in the previous studies enhanced the clustering performance of the standard DE algorithm. It is still necessary to provide a simple, efficient, and robust algorithm with fewer input parameters which assures to achieve a globally optimal solution. Therefore, this work proposes a DE variation for hard partitional clustering that provides better cluster solutions with faster convergence.

3. Differential evolution algorithm

Differential Evolution (DE) algorithm is one of the most often utilized evolutionary algorithms (EAs) for various complex real-world optimization problems. Like other EAs, DE keeps up a population of nominee solutions to the given problem. The initial population is composed of NP chromosomes that are randomly chosen solutions from the search space. A chromosome of the population at the gth generation is denoted as the vector $X_{i,g} = \{x_{i,g}^1, x_{i,g}^2, \dots, x_{i,g}^d\}$ where *d* is the feature of the problem, and *i*=1, 2,…, *NP*. Once the initial population is created, DE evolves the population generation by generation by performing three consecutive steps (mutation, crossover, and selection).

Inspired from biology, the mutation process of DE is a perturbation or alternation with a random component. In classical DE, the mutation operation is carried out with three different random chromosomes. Any two of the three chromosomes are used as donor vectors, and the rest one is used as a target vector. For each parent chromosome $X_{i,g}$ in the existing population, a trial vector is made by summing the scaled difference of donor vectors and the target vector as shown in Eq. (1).

$$V_{i,g} = X_{a,g} + f(X_{b,g} - X_{c,g}) \qquad (1)$$

Where $V_{i,g}$ is a trial vector, $X_{a,g}$ is a target vector, and $X_{b,g}$ and $X_{c,g}$ are donor vectors such that i is an integer within [1, NP], a, b, and c are random integers within [1, *NP*], and $i \neq a \neq b \neq c$, and then *f* is a scaled factor within (0,∞).

The crossover operation generates an offspring vector $U_{i,g}$ by recombining the recently created trial vector $V_{i,g}$ and the parent vector$X_{i,g}$. In classical DE, the binomial crossover is implemented as shown in Eq. (2).

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

Where $u_{i,g}^{j}$, $v_{i,g}^{j}$, and $x_{i,g}^{j}$ are the jth elements of $U_{i,g}$, $V_{i,g}$, and $X_{i,g}$ such that *i* an integers within [1, *NP*], *j* is an integer within [1, *d*], the crossover rate, *CR* ∈ (0,1), and rand(*j*) ∈ U(0,1).

The selection operation decides the survival vector $X_{i,g+1}$ for the next generation among the parent $X_{i,g}$ and offspring $U_{i,g}$ such that the offspring vector is selected if its obtained objective function value is better than this obtained by the parent; otherwise, the parent is taken for the subsequent generation. The deterministic selection is implemented as shown in Eq. (3).

$$X_{i,g+1} = \begin{cases} V_{i,g} & if\ f\left(V_{i,g}\right) > f(X_{i,g}) \\ X_{i,g} & otherwise \end{cases} \tag{3}$$

Where $f\left(U_{i,g}\right)$and$f(X_{i,g})$ are fitness values of the offspring and parent vectors.

3.1 Different mutation strategies in differential evolution

Different mutation strategies are applied to choose the target vector and to determine the number of differentials between donor vectors. DE/x/y notation is used to characterize the various strategies, in which x is the way to take the target vector, and y is the number of differentials. The random mutation strategy (DE/rand/1) mentioned in above is commonly utilized in the classical DE algorithm. The most commonly used mutation strategies [18] are as follow.

DE/best/1:

$$V_{i,g} = X_{best,g} + f\left(X_{a,g} - X_{b,g}\right) \tag{4}$$

DE/best/2:

$$V_{i,g} = X_{best,g} + f_1\left(X_{a,g} - X_{b,g}\right) + f_2\left(X_{c,g} - X_{d,g}\right) \tag{5}$$

DE/rand/2:

$$V_{i,g} = X_{a,g} + f\left(X_{b,g} - X_{c,g}\right) + f_2\left(X_{d,g} - X_{e,g}\right) \tag{6}$$

DE/current-to-rand/1:

$$V_{i,g} = X_{i,g} + f_1\left(X_{a,g} - X_{i,g}\right) + f_2\left(X_{b,g} - X_{c,g}\right) \tag{7}$$

DE/current-to-best/1:

$$V_{i,g} = X_{i,g} + f_1\left(X_{best,g} - X_{i,g}\right) + f_2\left(X_{a,g} - X_{b,g}\right) \tag{8}$$

Where *a*, *b*, *c*, *d*, and *e* are random integers within [1, *NP*] such that $i \neq a \neq b \neq c \neq d \neq e$, *f* is a scaled factor within (0,∞), and $X_{best,g}$ is the best chromosome in the existing population.

4. Quasi-opposition based learning

Opposition-based learning (OBL), introduced by Tizhoosh in 2005, was apioneering scheme for machine intelligence algorithms [19]. The core idea of OBL is exploring a better approximation of a number by regarding this number and its opposite one together. Several researches have conducted the incorporation of the OBL scheme in evolutionary algorithms to increase their search behaviors, accuracy, and convergence [20-23]. Quasi-opposition Based Learning (QBL) is an improved form of OBL which applies quasi-opposite points instead of opposite points. These points produced through QBL have more likelihood to be nearer to unknown solutions than the points created using OBL [24-25]. In this paper, the QBL scheme is utilized to generate fitter initial candidate solutions. The concept of opposite number and point described in [18] are as follow:

Definition 1 - Let *x* ∈ *[a, b]* be a real number. The opposite number $\breve{x}$ of *x* can be specified as follows:

$$\breve{x} = a + b - x \tag{9}$$

Definition 2 – Let *P*= $(x_1, x_2, \dots, x_d)$ is a point in *d*-dimensional space such that $(x_1, x_2, \dots, x_d) \in R$ and $x_i \in [a_i, b_i]$. Each element of the opposite point $\breve{P} = (\breve{x_1}, \breve{x_2}, \dots, \breve{x_d})$ can be defined as follows:

$$\breve{x}_\imath = \breve{a}_\imath + \breve{b}_\imath - x_i \tag{10}$$

The concept of quasi-opposite number and point described in [24] are as follows:

Definition 3 - Let $x \in [a, b]$ be a real number. The quasi-opposite number $\breve{x}^q$ is defined as follows:

$$\breve{x}^q = \begin{cases} rand(m, \breve{x}) & if\ \breve{x} \leq m \\ rand(\breve{x}, m) & otherwise \end{cases} \quad (11)$$

Where $m = \frac{a+b}{2}$.

Definition 4 - Let $P= (x_1, x_2, \ldots, x_d)$ is a point in d-dimensional space such that $(x_1, x_2, \ldots, x_d) \in R$ and $x_i \in [a_i, b_i]$. Each element of the opposite point $\breve{P}^q = (\breve{x}_1^q, \breve{x}_2^q, \ldots, \breve{x}_d^q)$ can be specified as follows:

$$\breve{x}_i^q = \begin{cases} rand(m_i, \breve{x}_i^q) & if\ \breve{x}_i^q \leq m_i \\ rand(\breve{x}_i^q, m_i) & otherwise \end{cases} \quad (12)$$

Where $m_i = \frac{a_i+b_i}{2}$.

Quasi-opposition based optimization creates a quasi-opposite chromosome for each candidate chromosome. By employing anobjective function, the fitness of each pair of chromosomes is measured. If the fitness of the quasi-opposite chromosome is superiorto this of the candidate chromosome, the candidate one is exchanged with the quasi-opposite one. Otherwise, the evolution process proceeds with the candidate one.

5. A clustering algorithm combining the modified DE algorithm with the QBL scheme

5.1 Solution representation

In order to utilize the differential evolution approach in data clustering, it is needed to represent each candidate cluster solution as a chromosome. This proposed approach uses the centroid-based representation that encodes the coordinates of cluster centers as a real-valued vector. The ith chromosome of the population is denoted as a vector $X_i = \{x_i^1, \ldots, x_i^d, \ldots, x_i^{(k-1)d+1}, \ldots, x_i^{kd}\}$, in which the first d elements denote the first cluster center; the next d elements denote the second one and so on. Each data object is allocated to the closest cluster based on a clustering criterion to carry out partitional clustering of a dataset. This criterion is employed as an objective (fitness) function to determine how fitting each chromosome in the population for a given clustering problem. The fitness of each chromosome is determined by totaling the distance between data instances and their respective cluster center as shown in Eq. (13).

$$f(X) = \sum_{j=1}^{k} \sum_{d \in C_j} D(d, c_j) \quad (13)$$

Where k represents the number of cluster, d is a data object in the given dataset, c_j is the center of the jth cluster C_j, and D (.) is the Euclidean distance between the data object and the center. The smaller the fitness value, the fitter the chromosome performs.

5.2 Population initialization

For the initialization of the DE population, random number generation is the typically used choice. In this paper, the QBL scheme is utilized to generate fitter initial candidate solutions. To obtain the initial population, k data objects are randomly selected from the given dataset, a vector is initialized with these data objects, and then a quasi-opposite of this vector is created. The fitter one from the pair of vectors is assigned to the DE initial population based on their fitness.

5.3 The proposed approach

In the proposed DE algorithm for clustering, a modified mutation strategy is employed to adjust the search behaviour of the DE algorithm. DE/best/1 strategy raises the convergence movement of the algorithm, but it is vulnerable to premature convergence. DE/rand/1 strategy conserves good diversity, but it diminishes the speed of the algorithm's convergence. The proposed mutation strategy consists of three random chromosomes and the best and current chromosomes. The best chromosome is used not only as a target vector but also as a donor vector, and the parent and three random chromosomes play as donor vectors. In other words, the modified mutation strategy uses the best one as a target vector, and three differentials from three pairs of donor vectors, such as one difference from the pair of parent and best chromosomes and the other two differences from three random vectors. The adoption of the guidance information of the best chromosome increases the convergence rate and the exploitation power of the algorithm, and the application of three differentials increases the diversity of the population and the exploration power of the algorithm. For each parent chromosome $X_{i,g}$ in the current population, the modified mutation strategy creates a trial vector $V_{i,g}$ as shown in Eq. (14).

$$V_{i,g} = X_{best,g} + f_1(X_{i,g} - X_{best,g}) + f_2(X_{a,g} - X_{b,g}) + f_3(X_{a,g} - X_{c,g}) \quad (14)$$

Where $X_{best,g}$ is the best chromosome of the gth generation, a, b, and c are random integers within [1, NP] such that $i \neq a \neq b \neq c$, f is a scaled factor within $(0, \infty)$. For crossover and selection operations, the proposed approach applies the binomial crossover and deterministic selection, respectively. The pseudo-code for the proposed clustering algorithm combining the modified DE algorithm with the QBL scheme (MDE-QBL) is given in Fig. 1.

```
Algorithm: MDE-QBL
Input: Dataset (D), Number of clusters (k), Number of population (NP), Scaling factors (f, f1, f2, f3), Crossover rate (CR)
Output: Optimal Cluster Solution
For i=1 to NP
    Initialize ith chromosome with k random data instances from D
    Create ith quasi-opposite chromosome
    Calculate the fitness of ith chromosome and its quasi-opposite chromosome
    Assign the fitter one to the initial population P
End
For i=1 to maxIt
    For i=1 to NP
        Generate ith trial vector by applying the proposed mutation strategy
        Generate ith offspring solution by applying the binomial crossover operator
    End
    For i=1 to NP
        Calculate the fitness of ith offspring solution
        Update the population by selection operation
    End
End
```

Figure. 1 A clustering algorithm combining the modified DE algorithm with the QBL scheme (MDE-QBL)

6. Experimental results

The experimental test was carried out on seven numerical datasets from the UCI machine learning repository, namely Iris, Wine, Thyroid, Breast Cancer, Pima, Glass, and Ecoli. The summary of these datasets is shown in Table. 1.

The achievement of the proposed algorithm was compared to the conventional DE algorithm with four of the most commonly used mutation strategies, namely DE/best/1, DE/rand/1, DE/current-to-rand/1, and DE/current-to-best/1. The control parameters were set based on [11] and [13] as follows: the number of maximum iterations, the number of population and crossover rate were set to 100, 100 and 0.9, respectively for all methods, the scaling factor was set to 0.5 for DE/best/1 and DE/rand/1, and all of the scaling factors for the rest others were set to 0.3. Each algorithm was separately tested 30 times on all of the selected datasets. The obtained objective function values by each algorithm are presented in Table 2 in terms of average and standard deviation.

The listed results in Table 2 showed that the proposed algorithm got better clustering results than others for all datasets. Moreover, it also achieved smaller standard deviation values than other methods for all of the used datasets, which pointed out that the proposed algorithm is more effective and stable than other DE variants.

The obtained objective function values throughout the iterations achieved by each competing method for all datasets are shown in Fig. 2. Although the proposed algorithm is slightly slower than DE/best/1 in early stages, it is able to search more extensively in later stages than all of the others, especially for Glass and Ecoli datasets. The quality of obtained clusters was also compared based on the sum of squared error (SSE) and the quantization errors. Table 3 and Table 4 presented the respective obtained results in terms of average and standard deviation. As shown in Table 3, the proposed algorithm achieved better SSE results than others except for Wine and Pima. Moreover, it got better values of quantization error than others except for Pima, as seen in Table 4. Specifically, the proposed algorithm achieved improved clustering performance in terms of solution quality and robustness with a faster convergence rate.

The performance of the proposed algorithm was also compared with the existing state of the art clustering techniques based on DE, such as a dynamic shuffled differential evolution algorithm for data clustering (DSDE), a forced strategy differential evolution algorithm for data clustering (FSDE), and a variance-based differential evolution algorithm with an optimal crossover for data clustering (VDEO). The obtained objective function values are presented in Table 5 in terms of average and standard deviation,

Table 1. Summary of datasets

Datasets	Number of data objects	Number of features	Number of clusters
Iris	150	4	3
Wine	178	13	3
Thyroid	215	5	3
Breast cancer	699	9	2
Pima	768	8	2
Glass	214	9	6
Ecoli	336	7	8

Table 2. Comparison of objective function values

Datasets	Index	DE/rand/1	DE/best/1	DE/current-to-rand/1	DE/current-to-best/1	MDE-QBL
Iris	Average	97.3809244	97.1257265	98.2257235	96.7280826	96.6554664
	Std.	0.28671121	0.28776993	0.2330539	0.067527291	3.09839E-06
Wine	Average	16293.9099	16316.9338	16306.5729	16296.246	16293.5275
	Std.	1.79819462	5.23245848	4.953413693	1.755794977	0.3282158
Thyroid	Average	1869.45039	1883.38591	1888.41288	1874.91641	1866.48303
	Std.	4.93738597	3.56456243	5.396856437	4.83995231	0.026051532
Breast Cancer	Average	2966.53782	3018.80216	2987.84142	2969.47028	2964.38984
	Std.	2.65086949	11.3991892	8.637445694	3.284382688	0.003417992
Pima	Average	47563.2065	47884.8688	47587.5049	47566.7832	47561.3404
	Std.	3.09224461	54.8016675	17.31487431	2.977584547	0.16293707
Glass	Average	224.339602	229.798054	244.030698	222.542509	214.790303
	Std.	7.93618395	2.58439611	2.155017781	4.262207958	1.32193952
Ecoli	Average	67.583721	67.9053545	72.1286431	66.9274105	63.7027324
	Std.	2.17513627	0.90432126	0.66660349	1.263528514	0.32627646

where all reported results except for the last column are directly obtained from [13]. As described in Table 5, the proposed algorithm is comparable to other competing algorithms and it obtained more robust solutions than others for almost all datasets.

7. Conclusion

This paper presented a DE based clustering algorithm. In the proposed algorithm, a new mutation strategy is introduced to increase the searchability of the DE algorithm, and QBL based population initialization technique is also applied to get fitter initial solutions. The proposed mutation strategy adopts the guidance information of the best chromosome to raise the exploitation power and convergence rate, and then, it applied three differentials to keep the diversity of the population and the exploration power of the algorithm. The exploitation and exploration ability of the algorithm are well balanced in this work. The conducted experiments on the seven real datasets indicated that the proposed algorithm outperforms the convention DE with four mutation strategies in term of their objective function values. It achieves superior quality and robust results with a faster convergence rate. Moreover, the proposed algorithm was also compared with the existing state of the art DE based clustering techniques on five datasets. The related results showed that the proposed algorithm is a comparable technique for hard partitional clustering. Further extension of this work is related to the implementation of the proposed algorithm on the distributed processing framework.

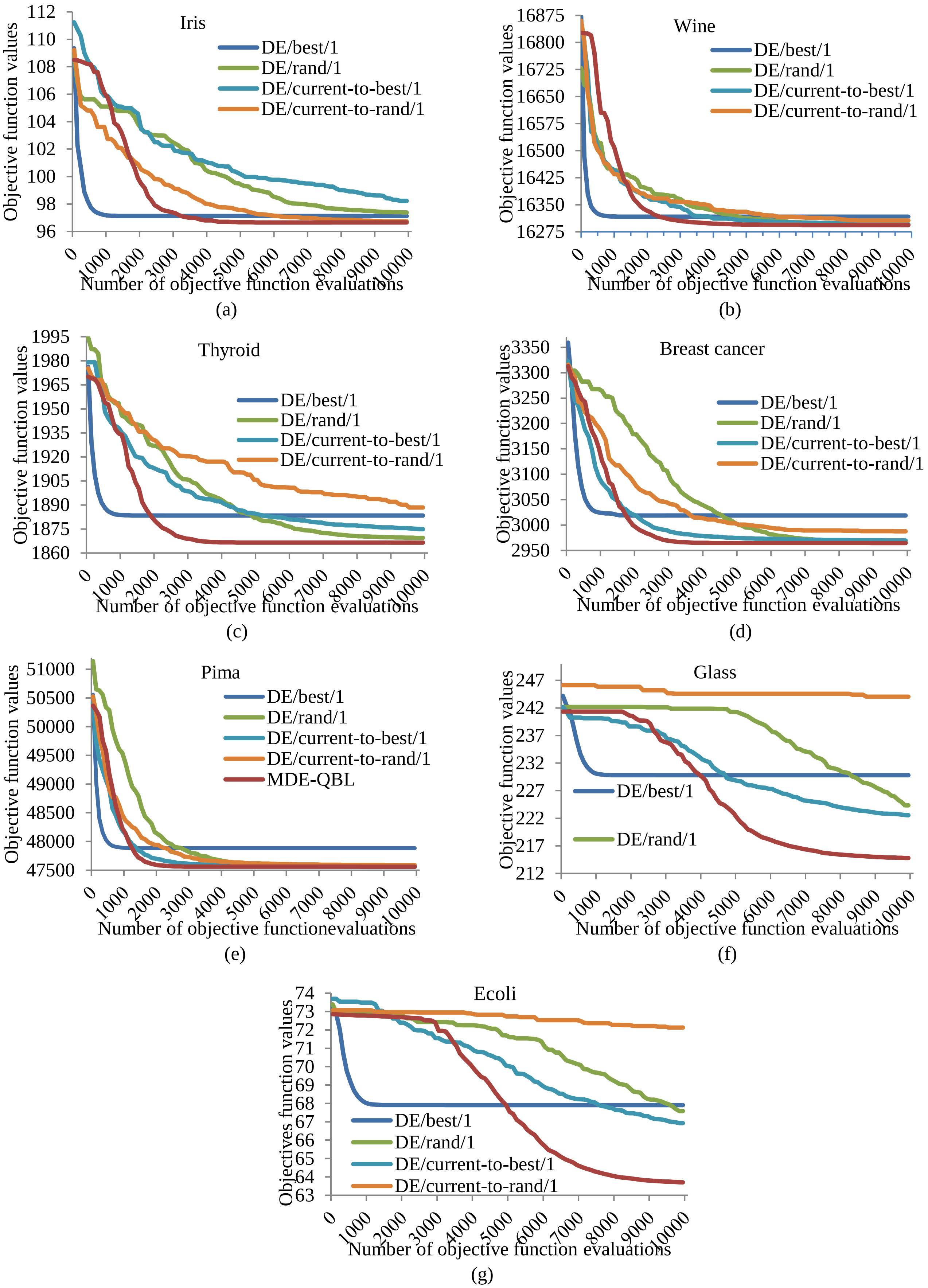
Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, P. P. W. C. and T. T. S. N.; methodology, P. P. W. C.; investigation, P. P. W. C.; writing—original draft preparation, P. P. W. C.; writing—review and editing, T. T. S. N.; supervision, T. T. S. N.

Received: June 5, 2020. Revised: July 27, 2020.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Figure. 2 Comparison of convergence performance on the selected dataset: (a) iris, (b) wine, (c) thyroid, (d) breast cancer, (e) pima, (f) glass, and (g) ecoli

Table 3. Comparison of sum of squared error

Datasets	Index	DE/rand/1	DE/best/1	DE/current-to-rand/1	DE/current-to-best/1	MDE-QBL
Iris	Average	82.0800965	80.9866801	83.3685664	80.3149833	80.136041
	Std.	0.94004685	0.65430367	1.488298053	0.154577131	0.001648966
Wine	Average	2564749	2549237.9	2580781.38	2581293.87	2585936.43
	Std.	27068.7995	31776.9966	32488.16905	21449.25797	16270.2647
Thyroid	Average	34884.4144	35391.341	35288.382	35201.0054	34793.6624
	Std.	239.216845	533.12527	537.5361905	237.0561105	111.3198744
Breast Cancer	Average	19503.1858	20163.7867	19871.8596	19568.6758	19457.6884
	Std.	58.336493	271.258444	169.3518486	84.81113703	1.775117536
Pima	Average	5876531.25	5909666.1	5916646.85	5878205.2	5877247.55
	Std.	1418.1087	57765.5027	28769.17591	2936.693797	765.3682647
Glass	Average	451.584795	504.460414	521.978414	467.785169	435.987293
	Std.	51.7028711	21.259169	34.68964865	32.75475979	40.9550011
Ecoli	Average	17.0615069	17.5507899	19.4565839	17.2151671	15.7089617
	Std.	1.06417957	0.73688293	0.552628073	0.597203844	0.412725648

Table 4. Comparison of quantization error

Datasets	Index	DE/rand/1	DE/best/1	DE/current-to-rand/1	DE/current-to-best/1	MDE-QBL
Iris	Average	0.648230088	0.647316521	0.654163175	0.644522413	0.64375
	Std.	0.002096196	0.003001606	0.002575488	0.00101929	1.3E-08
Wine	Average	95.7610459	96.0187704	95.8116934	95.668115	95.6275
	Std.	0.185679641	0.211108131	0.182229921	0.12394908	0.10682
Thyroid	Average	9.09637795	9.15417135	9.2730773	9.1844789	9.00613
	Std.	0.115034208	0.161775128	0.107299489	0.083235219	0.05606
Breast Cancer	Average	5.19391174	5.29663055	5.22435062	5.19876374	5.18876
	Std.	0.006918966	0.031849959	0.012082721	0.007111739	6.3E-06
Pima	Average	67.803803	68.247306	67.7693949	67.8087031	67.8006
	Std.	0.005570987	0.221579355	0.067872086	0.004526859	0.00023
Glass	Average	1.42850018	1.36619334	1.5327666	1.38235998	1.24923
	Std.	0.216891528	0.120531122	0.193121642	0.156474922	0.06684
Ecoli	Average	0.212012397	0.205757483	0.2186906	0.206791102	0.19406
	Std.	0.010604917	0.004651351	0.008642931	0.020067601	0.00332

Received: June 5, 2020. Revised: July 27, 2020.

Table 5. Comparison of objective function values

Datasets	DSDE	FSDE	VDEO	MDE-QBL
Iris	96.65 ± 0.00	96.70 ± 0.10	96.54 ± 0.00	96.65 ± 0.00
Wine	16292.28 ± 0.20	16325.26 ± 38.92	16293.56 ± 0.87	16293.52 ± 0.32
Thyroid	1874.00 ± 11.76	1882.60 ± 11.74	1867.51 ± 0.91	1866.48 ± 0.02
Breast Cancer	2968.28 ± 9.45	2964.47 ± 0.05	2964.43 ± 0.02	2964.38 ± 0.00
Glass	220.83 ± 12.16	245.02 ± 12.13	213.62± 1.99	214.79 ± 1.32

References

[1] O. Maimon and L. Rokach, *Data Mining and Knowledge Discovery Handbook*, 2nd ed., Springer, New York, 2010.

[2] R. Xu and D. Wunsch, “Survey of clustering algorithms”, *IEEE Transactions on Neural Networks*, Vol. 16, No. 3, pp.645-678, 2005.

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

[4] K. C. Wong, “A short survey on data clustering algorithms”, In: *Proc. of International Conf. on Soft Computing and Machine Intelligence*, Hong Kong, China, pp. 64-68, 2015.

[5] 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.

[6] S. Alam, G. Dobbie, Y. S. Koh, P. Riddle, and S. U. Rehman, “Research on particle swarm optimization based clustering: A systematic review of literature and techniques”, *Swarm and Evolutionary Computation*, Vol. 17, pp. 1-13, 2014.

[7] 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, Part C (Applications and Reviews)*, Vol. 39, No. 2, pp.133-155, 2009.

[8] 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.

[9] R. J. Kuo and F. E. Zulvia, “An improved differential evolution with cluster decomposition algorithm for automatic clustering”, *Soft Computing*, Vol. 23, No. 18, pp.8957-73, 2019.

[10] O. Tarkhaneh, J. Karimpour, S. Mazaheri, and E. Zamiri, “Automatic Clustering for Customer Segmentation by Adaptive Differential Evolution Algorithm”, In: *Proc. of Iranian Conf. on Signal Processing and Intelligent Systems (ICSPIS)*, Shahrood, Iran, pp. 1-9, 2019.

[11] 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, 2015.

[12] M. Ramadas, A. Abraham, and S. Kumar, “FSDE-Forced Strategy Differential Evolution used for data clustering”, *Journal of King Saud University-Computer and Information Sciences*, Vol. 31, No. 1, pp. 52-61, 2019.

[13] M. Alswaitti, M. Albughdadi, and N. A. M. Isa, “Variance-based differential evolution algorithm with an optional crossover for data clustering”, *Applied Soft Computing*, Vol. 80, pp. 1-7, 2019.

[14] I. Mishra I, I. Mishra, and J. Prakash, “Differential evolution with local search algorithms for data clustering: A comparative study”, In: *Proc. of Soft Computing: Theories and Applications*, Springer, Singapore, pp. 557-567,2019.

[15] M. A. Fitriani, A. Musdholifah, and S. Hartati, “Adaptive Unified Differential Evolution for Clustering”, *Indonesian Journal of Computing and Cybernetics Systems*, Vol. 12, No. 1, pp. 53-62, 2018.

[16] J. Tvrdík and I. Křivý, “Hybrid differential evolution algorithm for optimal clustering”, *Applied Soft Computing*, Vol. 35, pp. 502-512, 2015.

[17] G. Martinović and D. Bajer, Data Clustering with Differential Evolution Incorporating Macromutations”, In: *Proc. of International Conf. on Swarm, Evolutionary, and Memetic Computing*, Chennai, India, pp. 158-169, 2013.

[18] M. Leon and N. Xiong, “Investigation of mutation strategies in differential evolution for solving global optimization problems”, In: *Proc. of International Conf. on Artificial Intelligence and Soft Computing*, Zakopane, Poland, pp. 372-383, 2014.

[19] H. R. Tizhoosh, “Opposition-Based Learning: A New Scheme for Machine Intelligence”, In: *Proc. of International Conf. on Computational Intelligence for Modeling Control and Automation and International Conf. on Intelligent Agents, Web Technologies and*

Internet Commerce, Vienna, Austria, pp. 695-701, 2005.

[20] S. Rahnamayan, H. R. Tizhoosh, and M. M. Salama, "Oppositionbased differential evolution for optimization of noisy problems", In: *Proc. of IEEE International Conf. on Evolutionary Computation*, Vancouver, BC, Canada, pp. 1865-1872, 2006.

[21] S. Rahnamayan, H. R. Tizhoosh, and M. Salama, "A novel population initialization method for accelerating evolutionary algorithms", *Computers and Mathematics with Applications*, Vol. 53, No. 10, pp. 1605-1614, 2007.

[22] S. Rahnamayan, H. R. Tizhoosh, and M. Salama, "Opposition versus randomness in soft computing techniques", *Applied Soft Computing*, Vol. 8, No. 2, pp. 906-918, 2008.

[23] C. Yanguang, M. Zhang, and C. Hao, "A hybrid chaotic quantum evolutionary algorithm," In: *Proc. of IEEE International Conf. on Intelligent Computing and Intelligent Systems*, Xiamen, China, pp. 771-776, 2010.

[24] S. Rahnamayan, H. R. Tizhoosh, and M. M. Salama, "Quasioppositional differential evolution", In: *Proc. of IEEE Congress on Evolutionary Computation*, Singapore, pp. 2229-2236, 2007.

[25] B. Kazimipour, X. Li, and A. K. Qin, "A review of population initialization techniques for evolutionary algorithms", *In: Proc. of IEEE Congress on Evolutionary Computation*, Beijing, China, pp. 2585-2592, 2014.