Improved data clustering using multi-trial vector-based differential evolution with Gaussian crossover

Parham Hadikhani*
20h8561@ubd.edu.bn
School of Digital Science, Universiti Brunei Darussalam, Brunei

Daphne Teck Ching Lai
daphne.lai@ubd.edu.bn
School of Digital Science, Universiti Brunei Darussalam, Brunei

Wee-Hong Ong
weehong.ong@ubd.edu.bn
School of Digital Science, Universiti Brunei Darussalam, Brunei

Mohammad H.Nadimi-Shahraki
nadimi@iaun.ac.ir
Faculty of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran

ABSTRACT

Many algorithms have been proposed to solve the clustering problem. However, most of them lack a proper strategy to maintain a good balance between exploration and exploitation to prevent premature convergence. Multi-Trial Vector-based Differential Evolution (MTDE) is an improved differential evolution (DE) algorithm that is done by combining three strategies and distributing the population between these strategies to avoid getting stuck at a local optimum. In addition, it records inferior solutions to share information about visited regions with solutions of the next generations. In this paper, an Improved version of the Multi-Trial Vector-based Differential Evolution (IMTDE) algorithm is proposed and adapted for clustering data. The purpose here is to enhance the balance between the exploration and exploitation mechanisms in MTDE by employing Gaussian crossover and modifying the sub-population distribution between the strategies. To evaluate the performance of the proposed clustering, 19 datasets with different dimensions, shapes, and sizes were employed. The obtained results of IMTDE demonstrate improvement in MTDE performance by an average of 12 %. Our comparative study with state-of-the-art algorithms demonstrates the superiority of IMTDE in most of these datasets because of the effective search strategies and the sharing of previous experiences in generating more promising solutions. Source code is available on Github: https://github.com/parhamhadikhani/IMTDE-Clustering.

KEYWORDS

Differential evolution, Data clustering, Data mining, Optimization

1 INTRODUCTION

Clustering is a process that divides a set of data objects n with d dimensions into k separate groups [27]. Each division is called a cluster C_i. The members of each cluster are similar in terms of their characteristics, while the degree of similarity between different clusters is low. In such a case, the purpose of clustering is to detect the distinct groups and assign objects based on their similarity to the corresponding groups. The main difference between clustering and classification approaches is the lack of initial labels for observations [4]. In clustering, objects are grouped without prior knowledge, while classification methods use predefined classes in which objects are assigned. The application of clustering can be found in social network analyse [17], robotics [18], biology [35], and networks [16, 40, 41]. The types of clustering methods can be categorized as hierarchical clustering [20], mixed clustering [23], learning network clustering [36], and partition clustering [30]. The main goal of these clustering algorithms is to enhance the homogeneity and heterogeneity of clusters [19].

Clustering is considered a NP-hard problem [19]. Since there is a widespread view that evolutionary algorithms have good ability to find near-optimal solution to such problem [12], many evolutionary algorithms have been proposed for clustering such as classic meta-heuristic algorithms (simulated annealing [32],genetic algorithms [22], particle swarm optimization (PSO) [37], and differential evolution [6]). More recent methods include graph-based genetic programming [21], memetic elitist evolutionary algorithm based on decomposition [15], hybrid PSO with adopting Gaussian estimation and levy flight strategy [3] and density-based PSO [13]. DE [33] is a powerful algorithm for solving optimization problems in continuous space, proposed to overcome the main problem of genetic algorithm, namely the lack of local search. In recent years, various methods [1, 7, 14, 24] have been proposed to improve the performance of DE. Two important improvements are JADE [43] and CoDE [38], which modify the production of solutions in DE. Moreover, in [8] and [42], methods are provided to control the selection mechanism and find the optimal value of the parameters in DE. The selection of search strategies and the associated control parameter values are essential to improve the performance of DE. However, the use of only one strategy with uniform settings for different parameters to search for good solutions is not useful for improving DE. This motivated a combination of multiple DE search strategies to be applied, which can greatly improve DE performance. Multi-trial vector-based differential evolution (MTDE) is enriched by an efficient combination of search strategies, in which the search strategies and associated control parameter values are gradually trained from their previous experiences to generate more promising solutions. Instead of using one strategy as in DE, MTDE uses multiple search strategies to improve the search.

For these reasons, we adapt MTDE [25] to cluster data in this paper. Unlike MTDE, we employed Gaussian crossover in strategies R-TVP and G-TVP and modified the distributing policy to improve the exploitation and exploration abilities in our work. To better distribute the sub-population between the strategies and improve the balance between exploration and exploitation, the distributing

policy in MTDE is modified to allocate more populations for exploration in the early stages and to increase the number of population to exploit in the final stages over time. The aim of this research is to investigate the ability of the proposed Improved MTDE (IMTDE) to cluster data with different dimensions, shapes and sizes, and compare it with other algorithms.

The rest of paper is structured as follow. In section 2, IMTDE is reviewed. IMTDE clustering is explained in section 3. Experimental results and some concluding remark are provided in section 4 and 5, respectively.

2 MULTI-TRIAL VECTOR-BASED DIFFERENTIAL EVOLUTION ALGORITHM

MTDE, an improvement of DE algorithm, is one of the meta-heuristic and population-based algorithms for solving optimization problems. In DE, the population consists of a number of vectors, where each vector represents a potential solution to the optimization problem. The goal of DE is to generate a new solution for each target vector (current generation) that moves towards the optimal solution using a trial vector. A *trial vector* is an offspring obtained based on the crossover and mutation operations of two random vectors and a target vector. Since DE is highly dependent on the production of a trial vector, MTDE presents three strategies, consisting of representative based trial vector producer (R-TVP) to enhance the diversity of solutions, local random based trial vector producer (L-TVP) for proper balance of exploration and exploitation as well as rapid convergence, and global best history based trial vector producer (G-TVP) to escape from local optimum. In MTDE, inferior solutions that have information about visited areas are kept in a repository called *lifetime archive*. The lifetime archive is used to maintain population diversity and prevent premature convergence. The size of this repository is equal to the population of vectors. If the repository is full, the old vectors are replaced with new vectors. The MTDE algorithm begins with defining a sub-population distribution policy called "winner-based distributing" to distribute the population between the three trial vector production strategies. In each generation, the strategy that performs the best distributes the population according to its policy. In the first iteration, one of the strategies is selected as the best strategy by default. For later iterations, the strategy with the highest improvement rate in the previous generation compared to other strategies is selected as the best strategy. The improved rate of a strategy x is calculated from the Eq.(1).

$$IR_x = \frac{IV_x}{N_x} \tag{1}$$

IV_x is the number of improved vectors by strategy x and N_x is the number of populations allocated to strategy x. In MTDE, the time factor is ignored. In the early stages, the search space is explored to find the optimal region. To better distribute the sub-population between the strategies and improve the balance between exploration and exploitation, the distributing policy in MTDE (Eq.(1)) is modified to Eq.(2) in IMTDE to allocate more populations for exploration in the early stages and to increase the number of population to exploit in the final stages over time. Such treatments over time is not done originally.

$$IR_x = \begin{cases} \frac{\frac{IV_x}{MaxIter}}{N_x}, & \text{if } x \text{ is equal to L-TVP} \\ \frac{\frac{IV_x}{MaxIter-iter}}{N_x}, & \text{if } x \text{ is equal to R-TVP or G-TVP} \end{cases} \tag{2}$$

Where $MaxIter$ is the number of iterations, and $iter$ is the counter of iterations. After finding the improved rate and selecting the best strategy, vectors are distributed randomly between the sub-population of strategies according to the rules of encouragement and punishment defined in Eq.(3). In each generation, the population size of the best strategy (with the highest improved rate) is calculated based on N_{win} and the population size of the other two strategies is calculated based on N_{lose}.

$$N_x = \begin{cases} N_{win} = 0.6 \times N, N_{lose} = 0.2 \times N, & \text{if R-TVP or L-TVP have the highest } IR_x \\ N_{win} = 0.2 \times N, N_{lose} = 0.4 \times N, & \text{if G-TVP have the highest } IR_x \end{cases} \tag{3}$$

Where N is the number of vectors in the population.

In the *R-TVP*, the target vector x_i is moved based on a random vector in lifetime archive $x_{lifetime}$ and two vectors $x_{i_{best}}$ and $x_{i_{worst}}$ that have the best and worst fitness values respectively (Eq.(4)) among the N_{R-TVP} (Eq.(4)). The procedure of *G-TVP* (Eq.(7)) are similar to *R-TVP* except that in *G-TVP*, the global best vector x_{gb} is mutated based on two random vectors from population of G-TVP (N_{G-TVP}). To improve the ability of MTDE to exploit and deal with non-linear clusters, the Gaussian distribution is employed as a crossover operation in IMTDE for both strategies R-TVP and G-TVP. Trial vector (u_i) is obtained by using Gaussian-based crossover of two transformation matrices M and its binary inverse $\overline{M}$ with the x_i and the mutant vector in Eq.(5). M with $N \times D$ dimensions can be constructed from $M_{D\times D}$, which is a lower triangular matrix with the values of one, by replicating the square matrix $\frac{N}{D}$ times in $M_{N\times D}$. The remaining rows of $M_{N\times D}$ are filled with the initial rows of the square matrix.

Afterward, the rows of $M_{N\times D}$ are permuted randomly. $\overline{M}$ is obtained by replacing the inverse boolean value of each element in M.

$$\begin{aligned} v_{i_{R-TVP}} = x_i + f_i \times (x_{i_{best}} - x_i) + f_i \times (x_{i_{worst}} - x_i) \\ +\alpha_1 \times (x_{lifetime} - x_i) \end{aligned} \tag{4}$$

$$u_{i_{R-TVP}} = (x_i \times M + v_i \times \overline{M}) \times Gaussian(0, h) \tag{5}$$

$$v_{i_{G-TVP}} = x_{gb} + \alpha_2 \times (x_{r1} - x_{r2}) \tag{6}$$

$$u_{i_{G-TVP}} = (x_i \times M + v_i \times \overline{M}) \times Gaussian(0, h) \tag{7}$$

$$\alpha_1 = 2 - iter \times (\frac{2}{MaxIter}) \tag{8}$$

$$h_{i+1} = h_i - (\frac{1}{MaxIter}) \tag{9}$$

where f is a scale factor calculated by Cauchy distribution [34], h is the standard deviation of Gaussian distribution and according to Eq.(9) is linearly reduced on each iteration, α_1 is a coefficent computed by Eq.(8). In the strategy *L-TVP*, unlike two other strategies, the trial vector is obtained based on individual learning rather than evolution, hence it does not need a crossover and is calculated as follows:

$$v_{i_{L-TVP}} = x_i + f_i \times (x_{r1} - x_{r2}) + \alpha_2 \times (x_{lifetime} - x_i) \tag{10}$$

$$\alpha_2 = (initial - final) \times (\frac{MaxIter - iter}{MaxIter})^{\mu} \tag{11}$$

where x_{r1} and x_{r2} are two random vectors in the population of L-TVP (N_{L-TVP}), $x_{lifetime}$ is a random vector from lifetime archive, α_2 is a coefficient calculated by Eq.(11). In addition, *initial* and *final* are the initial and final values of the control parameter α_2 set by the user and, μ is the dimension of vectors. In the selection phase, trial vectors produced by each strategy are compared with their corresponding target vectors to update the population. If the trial vector has a better fitness value, it replaces the target vector and the target vector is stored in the lifetime archive. Otherwise, the target vector remains unchanged. Finally, after reaching the termination condition of iteration, the global vector x_{gb} is selected as the final optimal solution.

3 MTDE CLUSTERING

With respect to clustering, each vector in MTDE represents k cluster centroids and the structure of each vector x_i is shown below:

$$x_i = (C_{i1}, \ldots, C_{ij}, \cdots, C_{ik}) \tag{12}$$

Where C_{ij} represents the j^{th} cluster centroid in the i^{th} vector of cluster. Thus, the population size of vectors represents the number of possible clustering solutions for the desired data. Moreover, to evaluate the clustering quality of each vector, a fitness function called Sum of Square Error (SSE) is used. SSE is the sum of the squared differences between each data point and its cluster centroid. This function is used as a measure of variation within a cluster. SSE is calculated from the following equation.

$$SSE = \sum_{j=1}^{k} \sum_{\forall y \in C_j} \|y_i - C_j\|^2 \tag{13}$$

y_i is a data point belonging to the cluster C_j. Each vector is evaluated based on SSE at each iteration of clustering, where smaller values are favored. The cluster centroids in each vector are updated by one of the three strategies in the MTDE algorithm to improve clustering. The proposed clustering algorithm is shown in Algorithm 1. The MTDE clustering is illustrated in Fig. 1.

4 EXPERIMENTAL RESULTS

In this section, the performance of the proposed IMTDE clustering algorithm has been compared with twelve well-known algorithms that have been reported in existing literature, including K-Means [39], Affinity propagation [11], Mean-shift [5], Agglomerative clustering [20], DBSCAN [10], OPTICS [2], Gaussian mixtures model [31], BIRCH [44], Particle swarm optimisation (PSO) and Hybrid PSO and K-Means (HPSOkmeans) [10] and MTDE. IMTDE, MTDE, PSO and HPSOkmeans have been implemented in Python.

Algorithm 1: IMTDE Cluster Algorithm

Input: D={y_1,y_2,. . .,y_n} //Set of data points
k //Number of desired clusters
Output: Set of k clusters from global best vector

```
Initialize a population of vectors with random centroids in the search space
for iter=1 to the Maxiter do
    if iter==1 then
        Consider R-TVP as best strategy
    else
        Determine the best strategy in previous generation using Eq.(2)
    Distribute vectors using Eq.(3)
    for each strategy x (x can be R-TVP, L-TVP or G-TVP) do
        for x_i in N_x do
            Based on chosen strategy x, use Eq.(5),(6) or (7) accordingly for producing a trial vector
            Evaluate fitness value of x_i using Eq.(13)
            Update x_i
            Update lifetime archive
            Assign each data point y_i to the cluster with the nearest centroid based on euclidean distance
    Choose the vector with the best fitness value of all the vectors as the global best vector (x_gb)
Return x_gb
```

The Scikit-learn machine learning library has been used to implement other methods [29]. The comparison of results for each dataset is based on the best solution found in 30 different runs for each algorithm. The performance of the algorithms is compared based on criteria accuracy (AC), normalized mutual information (NMI) and adjusted rand index (ARI). The main parameters and their values in the IMTDE clustering algorithm are: *Maxiter*=200, *initial*=0.001, *final*=2, h=1 and number of vectors=200.

4.1 Dataset

For a comprehensive evaluation of the proposed clustering algorithm, 19 datasets were used. From the datasets used, six of them are artificial and the rest of datasets are real-world. The characteristics of these datasets are summarized in Table 1 [9].

4.2 Results and discussion

Fig. 2 and 3 showed the t-SNE biplots of IMTDE clustering performance on different datasets, with biplots of datasets IMTDE performed the best in Fig. 2 and those otherwise in Fig. 3. From Fig. 2, IMTDE has been able to identify clusters and distinguish overlapping between them very well in datasets with convex-shaped clusters. This is demonstrated in Aggregation (Fig. 2b), Digits (Fig. 2l), Iris (Fig. 2t), and Seeds (Fig. 2v) that the difference between the clusters has been well defined. From (Fig. 2h) and (Fig. 2r), despite the high dispersion of data points in dataset Glass and the high density of data points in Wine, IMTDE has been able to cluster

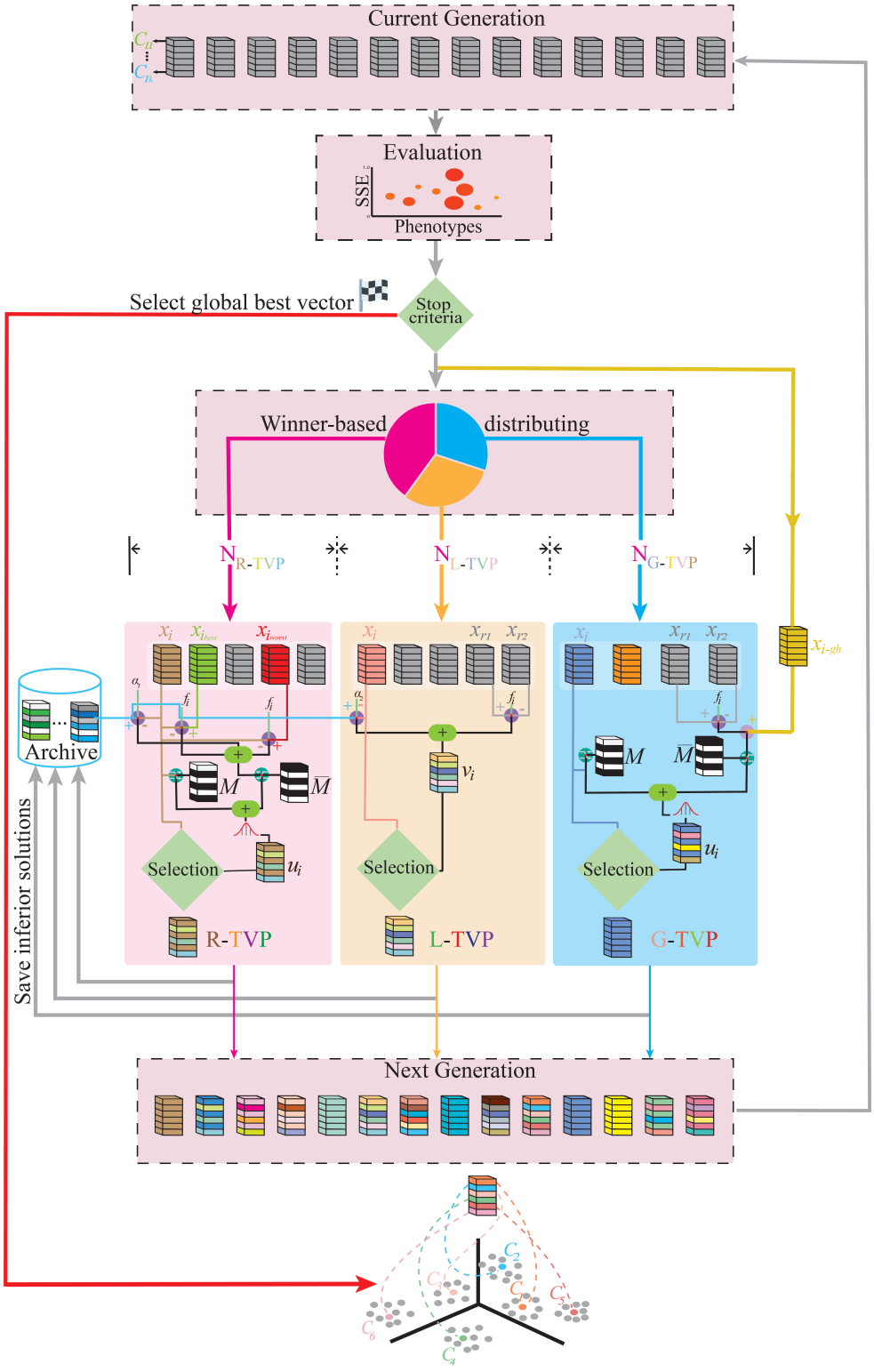
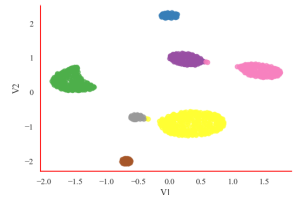
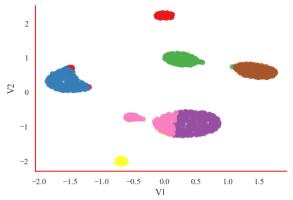


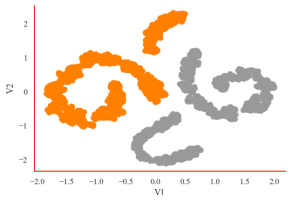
Figure 1: Illustration of proposed IMTDE clustering. First, vectors are initialized randomly. Fitness value of all vectors are then evaluated and global best vector is determined. Using winner-based distributing, vectors are distributed between three sub-populations randomly. Based on three strategy, trial vectors are produced in each sub-populations. The trial vectors produced by each strategy are evaluated to updated the population. To maintain the diversity of population and share the information about the visited areas, inferior solutions are kept in the lifetime archive at each generation. Finally, after reaching the stop criteria, the global best vector is selected as final clustering result.



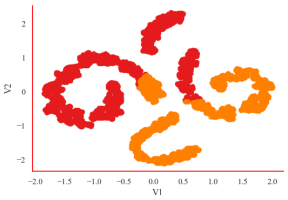
(a) Ground Truth of Aggregation



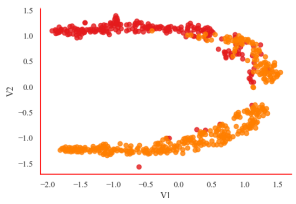
(b) IMTDE Results on Aggregation



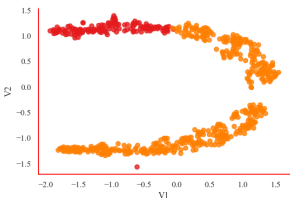
(c) Ground Truth of Banana



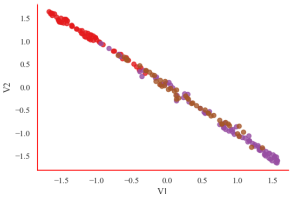
(d) IMTDE Results on Banana



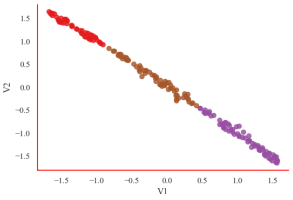
(e) Ground Truth of Breast cancer



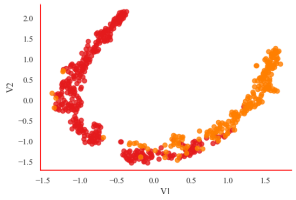
(f) IMTDE Results on Breast cancer



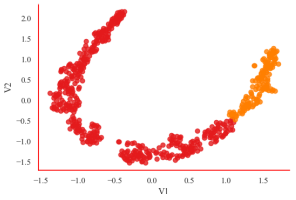
(g) Ground Truth of Wine



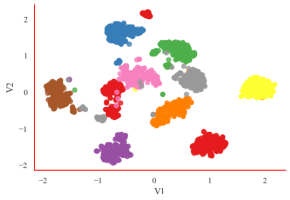
(h) IMTDE Results on Wine



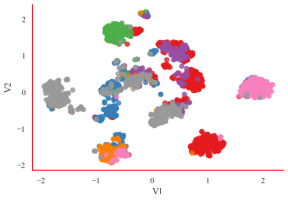
(i) Ground Truth of WDBC



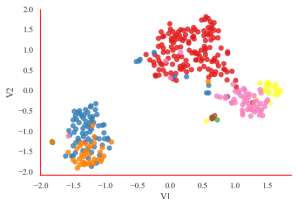
(j) IMTDE Results on WDBC



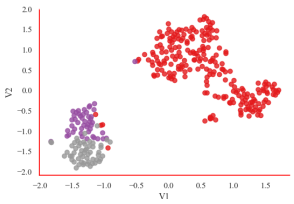
(k) Ground Truth of Digits



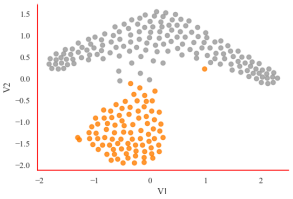
(l) IMTDE Results on Digits



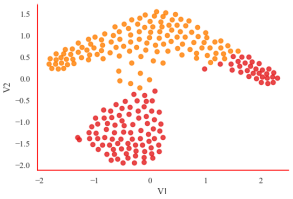
(m) Ground Truth of Ecoli



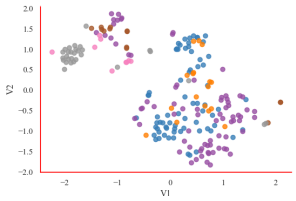
(n) IMTDE Results on Ecoli



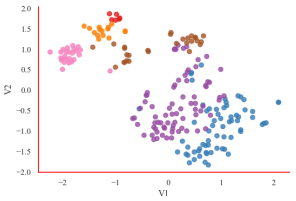
(o) Ground Truth of Flame



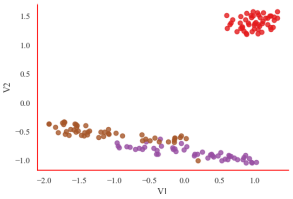
(p) IMTDE Results on Flame



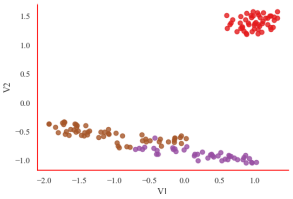
(q) Ground Truth of Glass



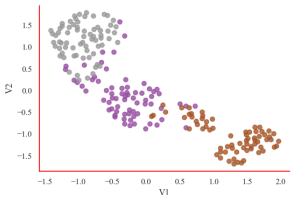
(r) IMTDE Results on Glass



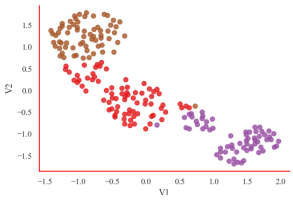
(s) Ground Truth of Iris



(t) IMTDE Results on Iris

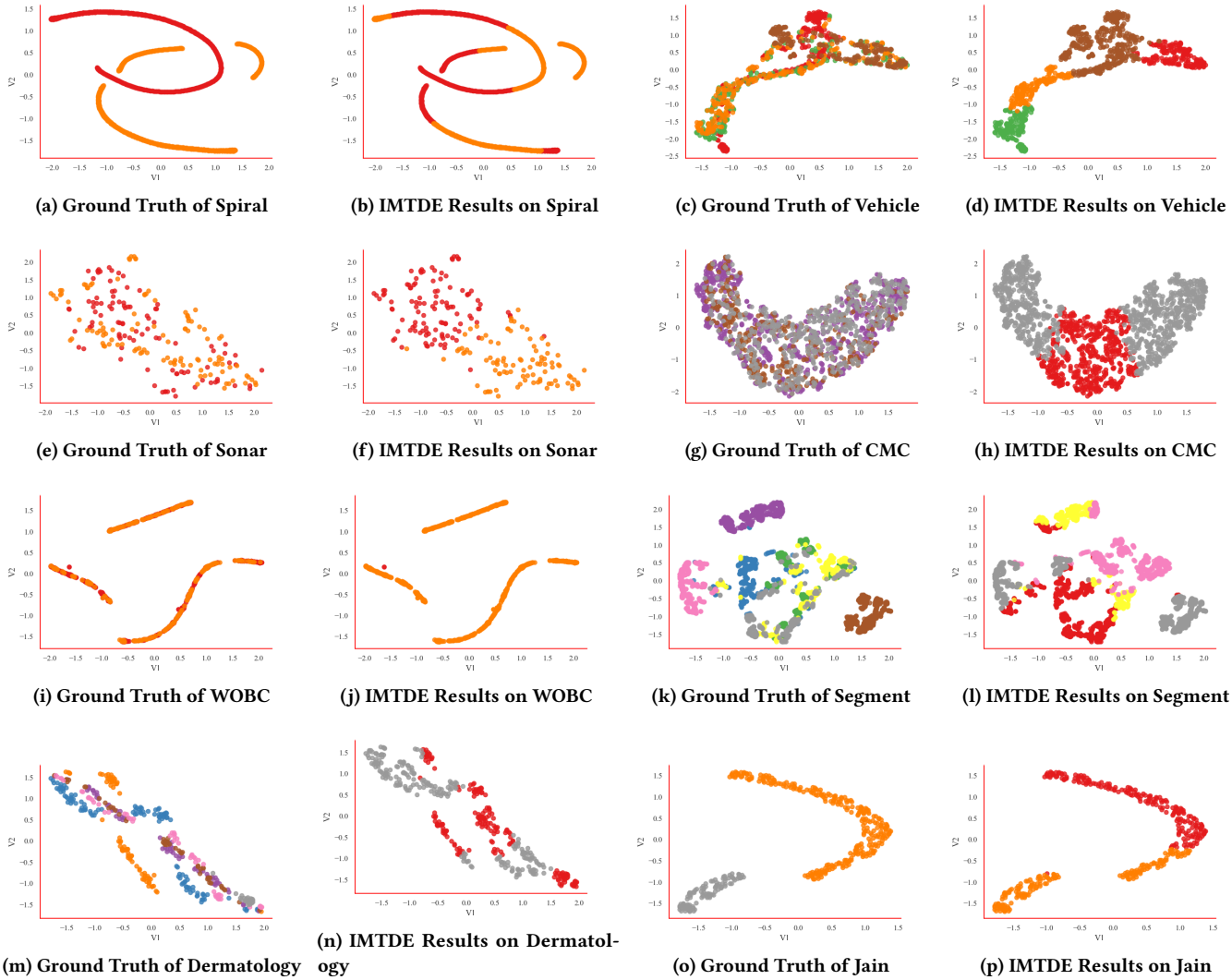


(u) Ground Truth of Seeds



(v) IMTDE Results on Seeds

Figure 2: T-SNE Biplots of IMTDE clustering on datasets that performed successfully.



(a) Ground Truth of Spiral (b) IMTDE Results on Spiral (c) Ground Truth of Vehicle (d) IMTDE Results on Vehicle

(e) Ground Truth of Sonar (f) IMTDE Results on Sonar (g) Ground Truth of CMC (h) IMTDE Results on CMC

(i) Ground Truth of WOBC (j) IMTDE Results on WOBC (k) Ground Truth of Segment (l) IMTDE Results on Segment

(m) Ground Truth of Dermatology (n) IMTDE Results on Dermatology (o) Ground Truth of Jain (p) IMTDE Results on Jain

Figure 3: T-SNE Biplots of IMTDE clustering on datasets that performed poorly

most data points correctly. This indicates that IMTDE is robust in the face of high data scatter and density. Moreover, IMTDE has been shown to work well in dealing with clusters that have a circular shape in challenging datasets Banana (Fig. 2d), Breast cancer (Fig. 2f), and WDBC (Fig. 2j). Most of the curved parts of these datasets are correctly identified by IMTDE. However, IMTDE has not been able to completely differentiate all the clusters in datasets such as WOBC (Fig. 3j) and Spiral (Fig. 3b) due to the use of the euclidean distance metric for clustering.

Fig. 4 shows a comparison of the worst, best, and average accuracy values between MTDE and IMTDE. The figure indicates IMTDE produced higher accuracy than MTDE In most datasets, which gave better outcomes than MTDE. On the one hand, this improvement is due to control employed on the population distribution over time that prevents early convergence and creates a balance between global and local search. On the other hand, in the datasets with high overlap and complexity, such as Spiral, Flame, Banana, Jain, and Digits, IMTDE has significantly improved performance MTDE. It is due to the Gaussian distribution in MTDE to map the original non-linear data into a higher-dimensional space in which they become separable and reduce overlap between data points. Thus, IMTDE not only performed better than MTDE but also increased the efficiency of MTDE in detecting non-linear clusters.

Fig. 5 shows the impact of the strategies used on convergence on dataset Dermatology. Using all three strategies together has resulted in a good convergence, avoiding local optimum trap, compared to the other combinations, with lowest SSE value at around iteration 200. The three strategies created appropriate population distributions for exploration and exploitation, maintaining a good balance between them.

Table 1: Characteristics of datasets

Datasets	Sample	Dimensions	Clusters	Type
Dermatology	366	34	6	Real-World
CMC	1473	9	3	Real-World
Sonar	208	60	2	Artificial
Vehicle	846	18	4	Real-World
Aggregation	788	2	7	Artificial
WDBC	568	30	2	Real-World
Glass	214	9	6	Real-World
Ecoli	335	7	8	Real-World
WOBC	698	10	2	Real-World
Seeds	210	7	3	Real-World
Segment	2310	18	7	Real-World
Wine	178	13	3	Real-World
Breast cancer	569	30	2	Real-World
Spiral	1000	2	2	Artificial
Banana	4811	2	2	Artificial
Flame	240	2	2	Artificial
Jain	373	2	2	Artificial
Digits	1797	64	10	Real-World
Iris	150	4	3	Real-World

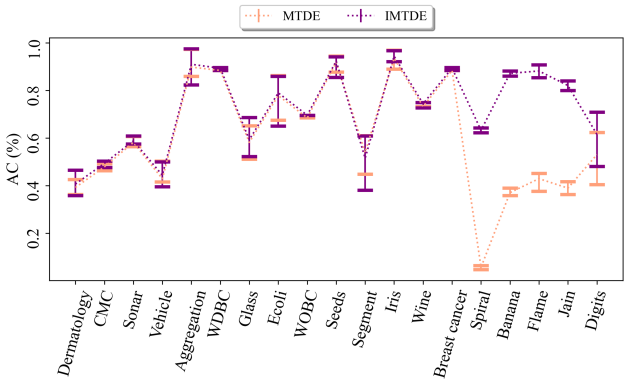


Figure 4: Comparison of the performance of IMTDE and MTDE based on the worst, best, and average accuracy for all datasets

Fig. 6 to 8 show the best performance based on AC, NMI, and ARI in all datasets. Table 3 shows the results of methods on datasets in which they outperformed others. Compared to other algorithms, IMTDE clustering was superior on ten datasets (CMC, Vehicle, Aggregation, Glass, Ecoli, Seeds, Segment, Iris, Banana, and Flame) in terms of AC. Based on NMI, IMTDE outperformed the rest on six datasets; CMC, Glass, Segment, Seeds, and Aggregation and based on ARI, performed best on eight datasets; CMC, Sonar, Vehicle, Glass, banana, Aggregation, Seeds, and Segment. IMTDE also outperformed DE due to the use of multiple strategies for producing trial vector. IMTDE maintains diverse solutions by using lifetime archive to keep inferior solutions, allowing the next generations to use past experiences and information from visited places to find the optimal solution.

The second best performing algorithm is the Gaussian mixture model (GMM) (see the purple dot in Fig. 6 to 8 and Table 3). This algorithm had the best performance in clustering based on all evaluation metrics on datasets Dermatology, WDBC, WOBC, Wine and Breast cancer. Because GMM uses the normal distribution function,

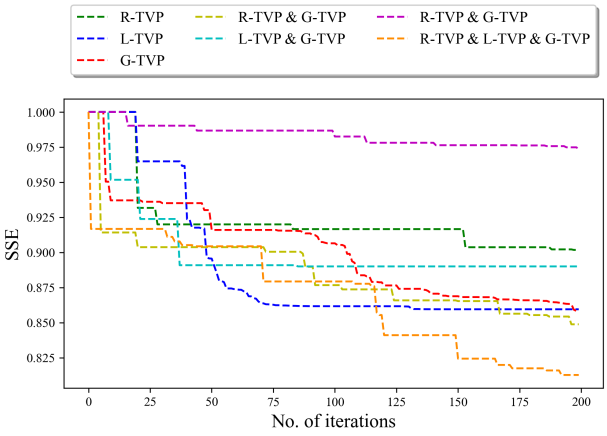


Figure 5: The impact of distribution policy and strategies on convergence on dataset Dermatology

it had a good performance in identifying and clustering elliptical shape data [28]. However, GMM uses expectation maximization (EM) algorithm to find clusters, which is subjected to EM problems of slow convergence and the local optimum trap when facing data with overlapping clusters [26]. Thus, it had a poorer performance than IMTDE. Algorithms Birch, Agglomerate, Optics and DE have also been able to surpass IMTDE in only one dataset. Other algorithms k-means, Affinity propagation, and Mean-shift had poor performances compared to others and did not outperform others in any datasets. Table 2 shows the results of methods on datasets in which they outperformed others.

Table 2: The algorithms on their best-performing dataset based on average values of AC, NMI, and ARI. IMTDE outperformed others on a majority of datasets.

Algorithms	AC	NMI	ARI
Birch	Banana 52.37 %		
Agglomerate	Jain 86.05 %	Jain 50.52 %	Jain 51.46 %
Optics	Spiral 100 %	Spiral 100 %	Spiral 100 %
PSO		Sonar 12.17 %	
		Banana 46.54 %	
DE	Sonar 61.53 %		Ecoli 74.84 %
HPSOK	Digits 79.18 %	Digits 74.45 %	Flame 60.74 %
		Ecoli 69.70 %	Digits 65.94 %
GMM	Dermatology 52.18 %	Dermatology 39.26 %	Dermatology 21.68 %
	WDBC 95.07 %	Vehicle 24.70 %	WDBC 81.12 %
	WOBC 92.12 %	WDBC 70.55 %	WOBC 70.86 %
	Wine 84.83 %	WOBC 66.49 %	Iris 90.38 %
	Breast cancer 95.07 %	Iris 89.96 %	Wine 60.74 %
		Wine 58.23 %	Breast cancer 81.16 %
		Breast cancer 70.61 %	
IMTDE	CMC 50.33,%	CMC 76.04 %	CMC 7.34 %
	Vehicle 49.96 %	Glass 48.67 %	Sonar 5.43 %
	Aggregation 97.50 %	Aggregation 93.48 %	Vehicle 16.76 %
	Glass 68.58 %	Seeds 76.47 %	Glass 32.53 %
	Ecoli 85.89 %	Segment 56.17 %	Banana 50.42 %
	Seeds 94.10 %	Flame 54.14 %	Seeds 76.32 %
	Segment 60.89 %		Segment 39.53 %
	Iris 96.76 %		Aggregation 92.10 %
	Banana 88.13 %		
	Flame 90.76 %		

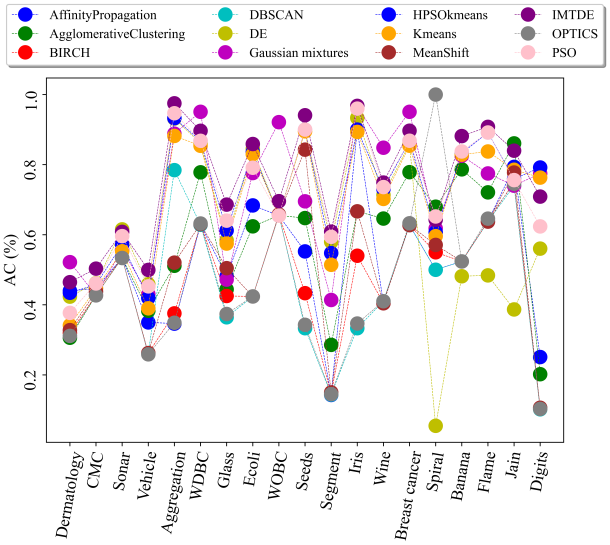


Figure 6: Best accuracy results of algorithms on 19 datasets

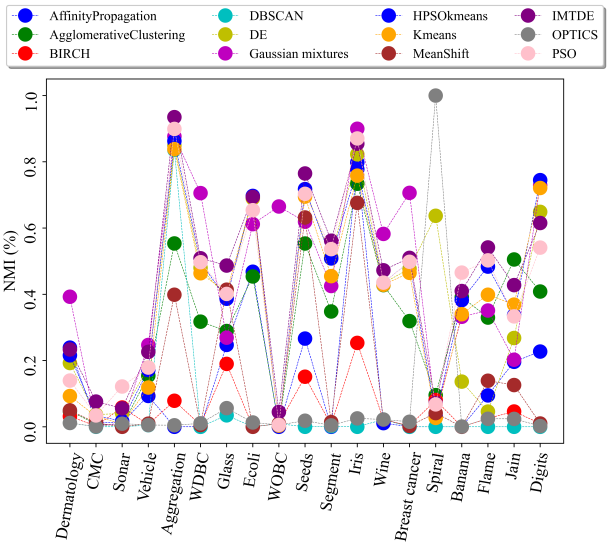


Figure 7: Best NMI results of algorithms on 19 datasets

Table 3 indicates a significant difference between the proposed and GMM, MTDE, and DE clustering methods using the Kruskal–Wallis test (p-value). Since the p-value of almost all of the datasets is less than 0.05 (significance level) with the 95 % confidence intervals for each median, we reject the null hypothesis, and conclude there is a significant difference between the proposed method with GMM, MTDE, and DE. In cases where the p-value is higher than the significant level (these cases are specified with* in Table 3), there is not enough evidence to reject the null hypothesis.

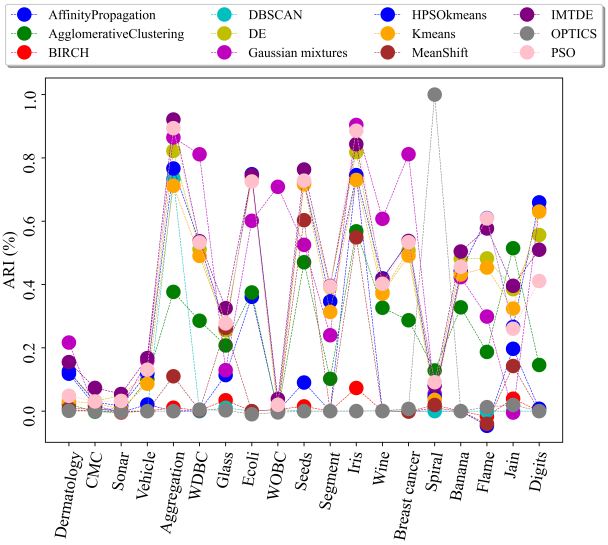


Figure 8: Best ARI results of algorithms on 19 datasets

Table 3: Comparison of Kruskal-Wallis test (p-value) between IMTDE and GMM, IMTDE and MTDE, and IMTDE and DE through 30 independent runs

Datasets	IMTDE vs GMM	IMTDE vs MTDE	IMTDE vs DE
Dermatology	0	0.01596	0.02559
CMC	0	0	0.00011
Sonar	0.63541*	0.2311*	0.29386*
Vehicle	0.343*	0.0057	0.00005
Aggregation	0.00444	0.92932*	0.00221
WDBC	0	0	0
Glass	0	0.01196	0.00071
Ecoli	0.15491*	0.00296	0.18824*
WOBC	0	0	0
Seeds	0	0.10707*	0.00015
Segment	0	0.05277*	0.01054
Wine	0	0.02658	0.92932 *
Breast cancer	0	0.01776	0
Spiral	0	0	0
Banana	0	0	0.03711
Flame	0	0	0.59456*
Jain	0	0	0.2675*
Digits	0	0	0.00927
Iris	0	0	0.00004

* p-value > 0.05: The differences between the medians are not statistically significant.

5 CONCLUSION AND FUTURE WORK

In this paper, an improved MTDE (IMTDE) is proposed and adapted to cluster data. IMTDE performance in clustering using time factor and Gaussian distribution improved by an average of 12 %. Comparing with other algorithms based on AC, NMI, and ARI evaluation, IMTDE has better clustering and a stronger ability to find the optimal global. Also, it was able to establish a good balance between exploration and exploitation. The results also showed that IMTDE has been able to enhance the results of DE due to the benefit of different search strategies for exploration and exploitation. One very important task that can improve the performance of IMTDE

clustering is to combine it with the algorithm GMM. This helps to properly cluster all the data with different shapes. Other possible tasks are to examine the performance of multi-objective clustering IMTDE. Automatically finding the optimal number of clusters using IMTDE clustering is another direction to pursue this work.

REFERENCES

[1] Ismail M Ali, Daryl Essam, and Kathryn Kasmarik. 2021. Novel binary differential evolution algorithm for knapsack problems. *Information Sciences* 542 (2021), 177–194.

[2] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod record* 28, 2 (1999), 49–60.

[3] Jianghui Cai, Huiling Wei, Haifeng Yang, and Xujun Zhao. 2020. A novel clustering algorithm based on DPC and PSO. *IEEE Access* 8 (2020), 88200–88214.

[4] Tanmoy Chakraborty. 2017. Ec3: Combining clustering and classification for ensemble learning. In *2017 IEEE international conference on data mining (ICDM)*. IEEE, 781–786.

[5] Yizong Cheng. 1995. Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence* 17, 8 (1995), 790–799.

[6] Swagatam Das, Ajith Abraham, and Amit Konar. 2007. Automatic clustering using an improved differential evolution algorithm. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans* 38, 1 (2007), 218–237.

[7] Wu Deng, Shifan Shang, Xing Cai, Huimin Zhao, Yingjie Song, and Junjie Xu. 2021. An improved differential evolution algorithm and its application in optimization problem. *Soft Computing* 25, 7 (2021), 5277–5298.

[8] Wu Deng, Junjie Xu, Yingjie Song, and Huimin Zhao. 2021. Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem. *Applied Soft Computing* 100 (2021), 106724.

[9] Dheeru Dua, Casey Graff, et al. 2017. UCI machine learning repository. (2017).

[10] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *kdd*, Vol. 96. 226–231.

[11] Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *science* 315, 5814 (2007), 972–976.

[12] B Galvan, D Greiner, J Périaux, M Sefrioui, and G Winter. 2003. Parallel Evolutionary Computation for solving complex CFD Optimization problems: a review and some nozzle applications. *Parallel Computational Fluid Dynamics 2002* (2003), 573–604.

[13] Hanjie Gao, Yintong Li, Petr Kabalyants, Hao Xu, and Rodrigo Martinez-Bejar. 2020. A novel hybrid PSO-K-means clustering algorithm using Gaussian estimation of distribution method and lévy flight. *IEEE access* 8 (2020), 122848–122863.

[14] Shangce Gao, Kaiyu Wang, Sichen Tao, Ting Jin, Hongwei Dai, and Jiujun Cheng. 2021. A state-of-the-art differential evolution algorithm for parameter estimation of solar photovoltaic models. *Energy Conversion and Management* 230 (2021), 113784.

[15] Germán González-Almagro, Alejandro Rosales-Pérez, Julián Luengo, José-Ramón Cano, and Salvador García. 2020. Improving constrained clustering via decomposition-based multiobjective optimization with memetic elitism. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. 333–341.

[16] Parham Hadikhani, Mohammadreza Eslaminejad, Meysam Yari, and Elmira Ashoor Mahani. 2020. An energy-aware and load balanced distributed geographic routing algorithm for wireless sensor networks with dynamic hole. *Wireless Networks* 26, 1 (2020), 507–519.

[17] Parham Hadikhani and Pooria Hadikhani. 2019. An adaptive hybrid algorithm for social networks to choose groups with independent members. *arXiv preprint arXiv:1910.01875* (2019).

[18] Parham Hadikhani, Daphne Teck Ching Lai, and Wee-Hong Ong. 2022. A Novel Skeleton-Based Human Activity Discovery Technique Using Particle Swarm Optimization with Gaussian Mutation. arXiv:2201.05314 [cs.CV]

[19] Eduardo Raul Hruschka, Ricardo JGB Campello, Alex A Freitas, et al. 2009. A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 39, 2 (2009), 133–155.

[20] Leonard Kaufman and Peter J Rousseeuw. 2009. *Finding groups in data: an introduction to cluster analysis*. Vol. 344. John Wiley & Sons.

[21] Andrew Lensen, Bing Xue, and Mengjie Zhang. 2017. GPGC: genetic programming for automatic clustering using a flexible non-hyper-spherical graph-based approach. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 449–456.

[22] Ujjwal Maulik and Sanghamitra Bandyopadhyay. 2000. Genetic algorithm-based clustering technique. *Pattern recognition* 33, 9 (2000), 1455–1465.

[23] J McLachlan Geoffrey. 1997. The EM algorithm and extensions/Geoffrey J. McLachlan, Thriyambakam Krishnan.

[24] Zhenyu Meng, Yuxin Zhong, and Cheng Yang. 2021. CS-DE: Cooperative Strategy based Differential Evolution with population diversity enhancement. *Information Sciences* 577 (2021), 663–696.

[25] Mohammad H Nadimi-Shahraki, Shokooh Taghian, Seyedali Mirjalili, and Hossam Faris. 2020. MTDE: An effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems. *Applied Soft Computing* 97 (2020), 106761.

[26] Iftekhar Naim and Daniel Gildea. 2012. Convergence of the em algorithm for gaussian mixtures with unbalanced mixing coefficients. *arXiv preprint arXiv:1206.6427* (2012).

[27] Charles Otto, Dayong Wang, and Anil K Jain. 2017. Clustering millions of faces by identity. *IEEE transactions on pattern analysis and machine intelligence* 40, 2 (2017), 289–303.

[28] Eva Patel and Dharmender Singh Kushwaha. 2020. Clustering cloud workloads: k-means vs gaussian mixture model. *Procedia Computer Science* 171 (2020), 158–167.

[29] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.

[30] Wang Peizhuang. 1983. Pattern recognition with fuzzy objective function algorithms (James C. Bezdek). *SIAM Rev.* 25, 3 (1983), 442.

[31] Carl Edward Rasmussen et al. 1999. The infinite Gaussian mixture model.. In *NIPS*, Vol. 12. Citeseer, 554–560.

[32] Shokri Z Selim and K1 Alsultan. 1991. A simulated annealing algorithm for the clustering problem. *Pattern recognition* 24, 10 (1991), 1003–1008.

[33] Rainer Storn and Kenneth Price. 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization* 11, 4 (1997), 341–359.

[34] Ryoji Tanabe and Alex Fukunaga. 2013. Success-history based parameter adaptation for differential evolution. In *2013 IEEE congress on evolutionary computation*. IEEE, 71–78.

[35] Nhi Tran, Younghoon Oh, Madeleine Sutherland, Qiang Cui, and Mei Hong. 2022. Cholesterol-Mediated Clustering of the HIV Fusion Protein gp41 in Lipid Bilayers. *Journal of molecular biology* 434, 2 (2022), 167345.

[36] Cheng-Fa Tsai, Han-Chang Wu, and Chun-Wei Tsai. 2002. A new data clustering approach for data mining in large databases. In *Proceedings International Symposium on Parallel Architectures, Algorithms and Networks. I-SPAN'02*. IEEE, 315–320.

[37] DW Van der Merwe and Andries Petrus Engelbrecht. 2003. Data clustering using particle swarm optimization. In *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, Vol. 1. IEEE, 215–220.

[38] Yong Wang, Zixing Cai, and Qingfu Zhang. 2011. Differential evolution with composite trial vector generation strategies and control parameters. *IEEE transactions on evolutionary computation* 15, 1 (2011), 55–66.

[39] Alan Williams. 1999. Calculating the global burden of disease: time for a strategic reappraisal? , 8 pages.

[40] Meysam Yari, Parham Hadikhani, and Zohreh Asgharzadeh. 2020. Energy-efficient topology to enhance the wireless sensor network lifetime using connectivity control. *Journal of Telecommunications and the Digital Economy* 8, 3 (2020), 68–84.

[41] Meysam Yari, Parham Hadikhani, Mohammad Yaghoubi, Raza Nowrozy, and Zohreh Asgharzadeh. 2021. An Energy Efficient Routing Algorithm for Wireless Sensor Networks Using Mobile Sensors. *arXiv preprint arXiv:2103.04119* (2021).

[42] Zhiqiang Zeng, Min Zhang, Tao Chen, and Zhiyong Hong. 2021. A new selection operator for differential evolution algorithm. *Knowledge-Based Systems* 226 (2021), 107150.

[43] Jingqiao Zhang and Arthur C Sanderson. 2009. JADE: adaptive differential evolution with optional external archive. *IEEE Transactions on evolutionary computation* 13, 5 (2009), 945–958.

[44] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. 1996. BIRCH: an efficient data clustering method for very large databases. *ACM sigmod record* 25, 2 (1996), 103–114.