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

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. Results show that IMTDE is superior to the compared algorithms in most 19 datasets used. The code is available here: https://github.com/parhamhadikhani/IMTDE-Clustering.

KEYWORDS

Differential evolution, Data clustering, Data mining, Optimization

ACM Reference Format:
Parham Hadikhani, Daphne Teck Ching Lai, Wee-Hong Ong, and Mohammad H.Nadimi-Shahraki. 2022. Improved data clustering using multi-trial vector-based differential evolution with Gaussian crossover. In *Genetic and Evolutionary Computation Conference Companion (GECCO '22 Companion), July 9–13, 2022, Boston, MA, USA.* ACM, Boston, USA, 4 pages. https://doi.org/10.1145/3520304.3528885

1 INTRODUCTION

Multi-trial vector-based differential evolution (MTDE) [2] is an improvement of Differential Evolution (DE) variant. 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. For these reasons, we adapt MTDE 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. 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.

In MTDE, inferior solutions that have information about visited areas are kept in a repository called *lifetime archive*. 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 strategies. 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.
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.

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

Where $MaxIter$ is number of iterations, and $iter$ is the counter of iterations.
The amount of dedicated population for each strategy is calculated

GECCO '22 Companion, July 9–13, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9268-6/22/07.
https://doi.org/10.1145/3520304.3528885

based on Eq.(3).

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

Where, the population size (N) of the best strategy is calculated based on N_{win} and the population size of the other two strategies are calculated based on N_{lose}.

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

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

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

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

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

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

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

where$f$ is a scale factor calculated by Cauchy distribution [4], 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:

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

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

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.

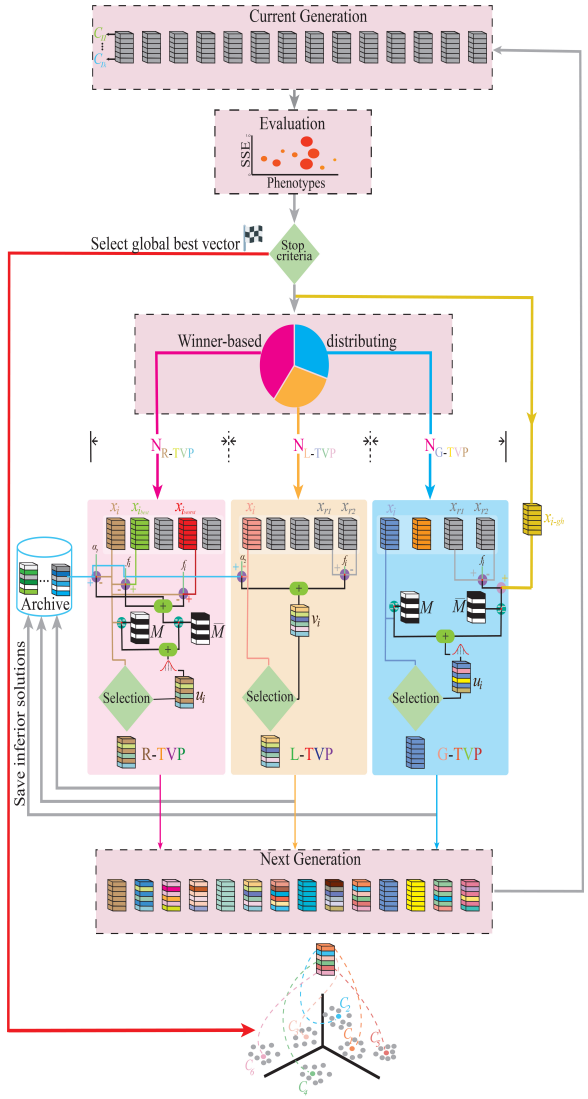

Figure 1: Illustration of IMTDE clustering. First, vectors are initialized randomly. The fitness value of all vectors is then evaluated and the global best is determined. Using modified winner-based distributing, vectors are distributed between three sub-populations randomly. Trial vectors are produced in each sub-populations. New generations are updated based on produced trial vectors. To maintain population diversity, inferior solutions are kept in the lifetime archive at each generation. Finally, the global best vector is selected as the final clustering result after reaching the stop criteria.

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.

2 IMTDE CLUSTERING

Each vector in IMTDE 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. Sum of Square Error (SSE) is used as the fitness function to evaluate the clustering quality of each vector. This function is used to measure variation within a cluster and is calculated as follows.

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

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.

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.(??)
    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
```

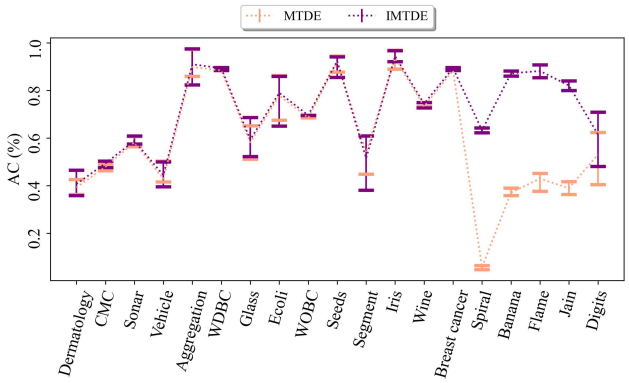

Figure 2: Comparison of the performance of IMTDE and MTDE based on the worst, best, and average accuracy for all datasets

2.1 Experiment

For a comprehensive evaluation of the proposed clustering algorithm, IMTDE has been compared with twelve well-known algorithms on 19 datasets [1]. The comparison of results for each dataset is based on the best solution found in 30 different runs for each algorithm. The main parameters and their values in IMTDE clustering algorithm are: *Maxiter*=200, *initial*=0.001, *final*=2, h=1 and number of vectors=200.

2.2 Results and Discussion

Fig. 2 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. 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. 3 shows the impact of the strategies used on convergence on dataset Dermatology. Using all three strategies together has resulted in a good convergence, avoiding a local optimum trap, compared to the other combinations, with the 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 2 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.

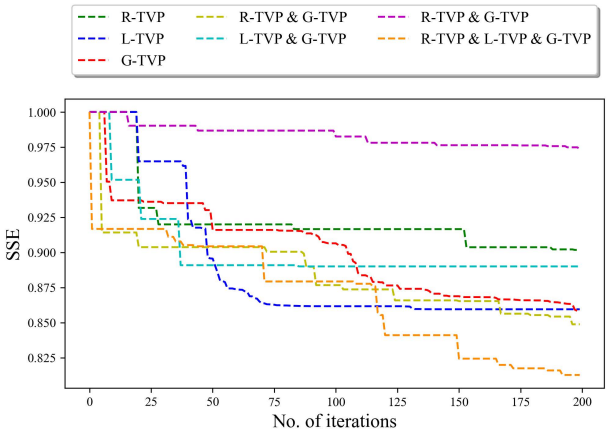

Figure 3: The impact of distribution policy and strategies on convergence on dataset Dermatology

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 multiple strategies for producing trial vector. IMTDE maintains diverse solutions by using lifetime archive to keep inferior solutions, allowing the next generations to use the past experiences and information from visited places to find the optimal solution. However, GMM uses the 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 [3]. Thus, it had a poorer performance than IMTDE. Table 2 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 2), there is not enough evidence to reject the null hypothesis.

Table 1: 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 %		

Table 2: 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.

3 CONCLUSION AND FUTURE WORK

In this paper, an improved MTDE (IMTDE) is proposed and adapted to cluster data. Comparing with other algorithms, IMTDE outperformed in most datasets because of stronger ability to find the optimal global and establishing a good balance between exploration and exploitation. One very important task that can improve the performance of IMTDE clustering is to combine it with the GMM algorithm. It helps to cluster all the data with different shapes properly. Other possible tasks are to examine the performance of multi-objective clustering IMTDE. Automatically finding the optimal number of clusters by using IMTDE clustering is another direction to pursue this work.

REFERENCES

[1] Dheeru Dua, Casey Graff, et al. 2017. UCI machine learning repository. (2017).
[2] 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.
[3] Iftekhar Naim and Daniel Gildea. 2012. Convergence of the em algorithm for gaussian mixtures with unbalanced mixing coefficients. *arXiv preprint arXiv:1206.6427* (2012).
[4] Ryoji Tanabe and Alex Fukunaga. 2013. Success-history based parameter adaptation for differential evolution. In *2013 IEEE congress on evolutionary computation*. IEEE, 71–78.