Information Dynamics and Applications
https://www.acadlore.com/journals/IDA

K-Means Clustering Algorithm Based on Improved Differential Evolution

Lei An[1], Xiaohua Sun[2*], Yan Wang[1]

[1] College of Artificial Intelligence, Baoding University, 071000 Baoding, China
[2] Department of Digital Media, Hebei Software Institute, 071000 Baoding, China

* Correspondence: Xiaohua Sun (fshw99@163.com)

Received: 07-25-2024 **Revised:** 08-28-2024 **Accepted:** 09-20-2024

Citation: L. An, X. H., Sun, and Y., Wang, "K-Means clustering algorithm based on improved differential evolution," *Inf. Dyn. Appl.*, vol. 3, no. 3, pp. 200–210, 2024. https://doi.org/10.56578/ida030305.

Abstract: The traditional K-means clustering algorithm has unstable clustering results and low efficiency due to the random selection of initial cluster centres. To address the limitations, an improved K-means clustering algorithm based on adaptive guided differential evolution (AGDE-KM) was proposed. First, adaptive operators were designed to enhance global search capability in the early stages and accelerate convergence in later stages. Second, a multi-mutation strategy with a weighted coefficient was introduced to leverage the advantages of different mutation strategies during various evolutionary phases, balancing global and local search capabilities and expediting convergence. Third, a Gaussian perturbation crossover operation was proposed based on the best individual in the current population, providing individuals with superior evolution directions while preserving population diversity across dimensions, thereby avoiding the local optima of the algorithm. The optimal solution output at the end of the algorithm implementation was used as the initial cluster centres, replacing the cluster centres randomly selected by the traditional K-means clustering algorithm. The proposed algorithm was evaluated on public datasets from the UCI repository, including Vowel, Iris, and Glass, as well as a synthetic dataset (Jcdx). The sum of squared errors (SSE) was reduced by 5.65%, 19.59%, 13.31%, and 6.1%, respectively, compared to traditional K-means. Additionally, clustering time was decreased by 83.03%, 81.33%, 77.47%, and 92.63%, respectively. Experimental results demonstrate that the proposed improved algorithm significantly enhances convergence speed and optimisation capability, significantly improving the clustering effectiveness, efficiency, and stability.

Keywords: K-means; Differential evolution algorithm; Clustering; Gaussian perturbation; Data mining; Cluster centre optimisation

1 Introduction

Clustering algorithms, as a type of unsupervised learning in data mining [1], are widely employed to uncover intrinsic associations and underlying patterns in data [2] without prior knowledge. By grouping data based on characteristics such as "similarity" or "proximity," clustering enables the categorisation of data into distinct classes [3]. However, different algorithms or varying parameter settings within the same algorithm often lead to divergent data categorisation or reveal different clustering structures [4]. Current clustering algorithms can be broadly categorised into partition-based methods [5], such as K-means and k-medoids, density-based methods [6], such as Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Ordering Points To Identify the Clustering Structure (OPTICS), and grid-based methods [7], such as WaveCluster and Statistical Information Grid (STING). Among these, the K-means clustering algorithm, proposed by Dubey and Choubey [8], has been extensively applied in data mining and knowledge discovery due to its simple calculation and low linear time complexity [9]. Despite these advantages, the traditional K-means clustering algorithm relies on the random selection of initial cluster centres [7]. This randomness [10] can cause the initial cluster centres to deviate significantly from the dataset or become overly concentrated. As a result, clustering outcomes may be compromised, which reduces the accuracy of clustering and increases computation time and resource consumption, thereby affecting the effectiveness of clustering [11].

To address these limitations of the traditional K-means clustering algorithm [12] and improve both the effectiveness and efficiency of clustering, significant efforts have been made to refine the k-means clustering algorithm. For instance, Bai et al. [13] improved the artificial fish swarm algorithm by dynamically updating the rotation angle

of quantum rotation gates and replacing the mutation strategy from NOT gates to Hadamard gates. This improved artificial fish swarm algorithm was used to select initial cluster centres for K-means, enhancing both convergence speed and clustering performance. Sun et al. [14] improved the velocity inertia weight and position update formula of the particle swarm optimisation algorithm and incorporated attribute weights into the traditional Euclidean distance metric. The improved particle swarm algorithm was used to optimise the K-means, effectively enhancing clustering performance and reducing the number of iterations required. Wang et al. [15] addressed the slow convergence and lack of diversity in the cuckoo optimisation algorithm by integrating the principles of particle swarm optimisation and applying the improved algorithm to K-means, achieving higher accuracy and greater algorithmic stability. Similarly, Wang et al. [16] employed multivariate statistical distance based on sample variance and introduced an improved artificial bee colony algorithm to determine initial cluster centres, effectively mitigating the susceptibility of K-means to local optima. Chen et al. [17] introduced a weighted Euclidean distance into the firefly optimisation algorithm, optimising the selection of initial cluster centres and reducing the impact of uncertain factors such as outliers, thereby improving both clustering performance and convergence speed. Hu et al. [18] incorporated the lion optimisation algorithm into K-means, using the optimal solution of the lion king at the end of the algorithm as the cluster centres. This approach effectively reduced the dependency of K-means on the initial cluster centre selection. In summary, leveraging intelligent optimisation algorithms to refine the selection of initial cluster centres has been shown to significantly enhance the clustering performance of the K-means clustering algorithm.

Differential evolution has been recognised as an efficient global optimisation algorithm, demonstrating robust performance in handling data while requiring relatively few parameter settings. Its simplicity in implementation and adjustment [19] further enhances its appeal. However, similar to many evolutionary algorithms, differential evolution also faces challenges such as reliance on a single evolutionary strategy, susceptibility to local optima, and slow convergence. To address these limitations in differential evolution and the K-means clustering algorithm, an improved K-means clustering algorithm based on enhanced differential evolution was proposed and its improvement effect on K-means was discussed. In contrast to traditional differential evolution, the proposed method introduces adaptive operators capable of dynamically adjusting based on the current evolutionary stage. This mechanism enables a broad exploration of the solution space during early stages while focusing on fine-tuning high-quality solutions in later stages, thereby significantly improving convergence speed. Additionally, a multi-mutation strategy based on a weighted coefficient was developed. This coefficient adjusts with the number of evolutionary iterations, allowing different mutation strategies to be employed during different evolutionary phases. This design achieves a balance between global and local search capabilities, reducing the risk of local optima caused by a single mutation strategy and further accelerating convergence. Furthermore, a Gaussian perturbation crossover operation guided by the best individual in the current population was designed. This approach not only directs individuals towards superior evolutionary directions but also maintains diversity across dimensions of the population, further reducing the likelihood of stagnation in local optima. The optimal solution generated at the termination of the improved differential evolution was employed to replace the randomly selected initial cluster centres of traditional K-means. Comparative experiments conducted using public datasets from the UCI repository and the synthetic dataset indicate that the proposed algorithm effectively reduces the dependency of K-means on the selection of initial cluster centres and captures the intrinsic structure of data more accurately, thereby enhancing clustering quality and efficiency.

2 K-Means Clustering Algorithm Based on the Improved Differential Evolution

2.1 Improved Differential Evolution Algorithm

2.1.1 Adaptive operators

In the differential evolution algorithm, mutation and crossover operations constitute the central mechanisms of the algorithm. The mutation factor (F) and crossover factor (C_R) play critical roles in the algorithm. In traditional differential evolution, these factors are set as fixed values, which limits the algorithm's optimisation capability and convergence efficiency, thereby impeding its performance enhancement [20]. During the early stages of evolution, population diversity should be maintained to enhance global optimisation capability. In contrast, during the middle and later stages, the focus should shift towards strengthening local search capability to accelerate convergence.

To enalbe F and C_R to meet the varying optimisation requirements across different evolutionary stages, adaptive operators were employed in this study to dynamically balance global and local search capabilities as the algorithm evolves. The adaptive strategy for the operators is defined as follows:

$$F = F_{max} - (F_{max} - F_{min}) (G/G_{max})^2 \tag{1}$$

$$C_R = C_{Rmax} - (C_{Rmax} - C_{Rmin}) (G/G_{max})^2 \tag{2}$$

where, F_{min} and F_{max} represent the lower and upper bounds of F, set to 0.3 and 0.9, respectively; C_{Rmin} and C_{Rmax} denote the lower and upper bounds of C_R, also set to 0.3 and 0.9, respectively; G represents the number of iterations for the current algorithm; and G_{max} indicates the maximum number of iterations of the algorithm.

The above equations enable the adaptive adjustment of F and C_R, ensuring a linear decrease in both factors as the algorithm progresses. This allows the algorithm to possess a larger search space and stronger global optimisation capability during the early stages of evolution, while exhibiting enhanced local search capability in the middle and later stages. Consequently, convergence speed of the algorithm can be improved.

2.1.2 Multi-mutation strategy with a weighted coefficient

To enhance the optimisation capability of the differential evolution algorithm, maintaining population diversity during the early stages of evolution is critical for improving global optimisation capability. The DE/rand/1 mutation strategy is characterised by its wide optimisation range, making it well-suited for global search in the early stages of the algorithm. In contrast, during the middle and later stages of evolution, global search has already been performed, and the focus should shift to local optimisation around the best individual and its vicinity. The DE/current-to-best/2 mutation strategy is more effective during these stages due to its strong local search capability and faster optimisation speed.

Based on the characteristics of different mutation strategies, the DE/rand/1 and DE/current-to-best/2 mutation strategies were combined into an improved differential evolution algorithm based on the multi-mutation strategy. An adaptive weight coefficient W was introduced to dynamically adjust the proportion of each strategy throughout the evolutionary process. This enables different mutation strategies to dominate at different stages of the algorithm, maximising their respective advantages.

$$W = W_{min} + (W_{max} - W_{min})\,(G/G_{max}) \tag{3}$$

$$\begin{aligned} V_{i,G+1} =& (1-W)\,[X_{r1,G} + F\,(X_{r2,G} - X_{r3,G})] \\ &+ W\,[X_{i,G} + F\,(X_{best,G} - X_{i,G}) \\ &+ F\,(X_{r4,G} - X_{r5,G}) + F\,(X_{r6,G} - X_{r7,G})] \end{aligned} \tag{4}$$

where, W is the weight factor, ranging from 0 to 1; G is the current iteration number; G_{max} is the maximum number of iterations; $V_{i,G+1}$ denotes the mutated individual in generation $G+1$; $X_{best,G}$ represents the individual with the best fitness value in the current population; and $r_1, r_2, r_3, r_4, r_5, r_6, r_7$ represents 7 random numbers that are not equal to i or mutually different from each other.

The weight coefficient W changed with the number of iterations, dynamically balancing the proportion of two mutation strategies in different evolution stages of the algorithm. During the early stages of evolution, the value of W remained small, allowing the DE/rand/1 mutation strategy to dominate. This facilitated global optimisation across a larger search space. As the algorithm progressed to the middle and later stages, the value of W increased, enabling the DE/current-to-best/2 strategy to take precedence. This transition improved the local search capability of the algorithm, guided individuals towards better evolutionary directions, and accelerated the convergence of the algorithm.

2.1.3 Gaussian perturbation crossover operation

In the differential evolution algorithm, mutation and crossover operations were used to explore and exploit new solution spaces. Although the mutation and crossover factors can be adaptively adjusted, these operations typically occur along the individual "dimensions." If the values of the population in a certain dimension converge towards a fixed value or become nearly identical, the differential evolution algorithm loses its optimisation capability in that dimension. To maintain diversity across dimensions and avoid the local optima of the algorithm, a Gaussian perturbation mechanism based on the best individual in the current population was introduced into the crossover operation. This approach leverages the best individual to guide the evolution of others, while Gaussian perturbation probabilistically generates new values in each dimension, preserving diversity of the population across dimensions. The specific steps are as follows:

Step 1: For a given dimension n in a clustering problem, a random number $rand(0,1)$ was taken. If $rand(0,1)$, then $U^n_{i,G} = V^n_{i,G}$ and the operation for that dimension can be terminated. Otherwise, proceed to Step 2.

Step 2: Another random number $rand(0,1)$ was taken. If $rand(0,1) \leq 0.7$, then $U^n_{i,G} = X^n_{best,G}$. Otherwise, a new random value in the dimension can be generated using Gaussian perturbation, as defined by:

$$U^n_{i,G} = X^n_{best,G} \times [1 + C \times N(0,1)] \tag{5}$$

where, C is the control parameter for Gaussian perturbation, set to 0.1; and $N(0,1)$ represents a random value sampled from a normal distribution with a mean of 0 and a standard deviation of 1.

The new crossover operation encouraged new individuals to move towards the individual with the best fitness value in the current population. At the same time, Gaussian perturbation was used to prevent the population from tending towards a fixed value in a certain dimension, thereby losing optimisation capability in the dimension. This prevents the algorithm from stagnating in local optima. Combined with the previously described adaptive operators C_R, this method allows global optimisation capability to be retained during the early stages of evolution. In the later stages, it enhances local search capability and accelerates convergence of the algorithm.

2.2 K-Means Clustering Algorithm Based on the Improved Differential Evolution

The improved differential evolution algorithm was integrated with the K-means clustering algorithm. The fitness function defined in Eq. (6) was employed to evaluate the fitness value of each individual. The improved differential evolution algorithm was used for iterative optimisation, and the optimal individual obtained at the end of the algorithm was used to replace the randomly selected initial cluster centres of the traditional K-means clustering algorithm.

2.2.1 Fitness function

In clustering algorithms, the final objective is to minimise the intra-cluster compactness while maximising the inter-cluster compactness. Therefore, the SSE [21] was adopted as the fitness function for the improved differential evolution algorithm:

$$fitness = SSE = \sum_{i,k} (x_{i,k} - c_k)^2 \tag{6}$$

where, $x_{i,k}$ represents a data point; and c_k denotes the centre of the cluster to which the data point belongs. A smaller SSE indicates better clustering performance, while a larger SSE suggests poorer clustering performance.

2.2.2 Algorithm workflow

a) The population size NP was set to ten times the solution dimension D. The mutation factor (F) and crossover factor (C_R) were adaptively taken according to Eqs. (1) and (2), with $F_{min} = 0.3$, $F_{max} = 0.9$, $C_{Rmin} = 0.3$, and $C_{Rmax} = 0.9$. With G as the current evolutionary iteration number, the maximum evolution number (G_{max}) of the algorithm was set to 1200 and the Gaussian perturbation coefficient (C) was set to 0.1.

b) The population of size NP was initialised.

c) The improved mutation operation, as described in Section 2.1.2, was applied to individuals $X_{i,G}$ in the population, generating mutant individuals $V_{i,G}$.

d) The improved crossover operation, as described in Section 2.1.3, was applied to both parent and mutant individuals, denoted as $X_{i,G}$ and $V_{i,G}$, to produce intermediate trial individuals $U_{i,G}$.

e) The fitness values of both the parent and intermediate trial individuals, denoted as $X_{i,G}$ and $U_{i,G}$, were computed and compared. Individuals with better fitness values were selected to form the next generation population for further evolution. Steps 3, 4, and 5 were repeated iteratively until the maximum number of evolutionary iterations was reached.

f) The optimal individual output by the algorithm was utilised as the initial cluster centre for the K-means clustering algorithm. The clustering process iterated until the stopping criterion or the maximum number of clustering iterations was met, after which the clustering results were output.

The pseudocode for the K-means clustering algorithm based on the improved differential evolution is presented as follows:

Function ImprovedDifferentialEvolution(Dimensionality):
  //InitialisePopulationP
  P = *InitialisePopulation(N, Dimensionality)*
  EvaluateFitness(P)
//EvolutionaryProcess
  for G from 1 to Gmax do:
    //CalculateMutationFactor F, CrossoverFactorCR, WeightCoefficientW
    $F = Fmax - (Fmax - Fmin)^*(G/Gmax)^2$
    $CR = CRmax - (CRmax - CRmin) * (G/Gmax)^2$
    W = *Wmin +(Wmax-Wmin* $) * (G/Gmax)^2$ // *ForEachIndividual* (i)
    for i from 1 to N do:
      //GenerateMutantIndividual(Vi)
      $r1, r2, r3, r4, r5, r6, r7$ = *SelectRandomIndividuals*(P)
      best = $SelectIndividuals(P)$
      Vi = GeneratedFromEquation(4)
      //PerformCrossoverOperationToGenerateIntermediateTrialIndividual(Ui)
      Ui = CreateEmptyIndividual()

for j *from 1 to Dimensionality do:*
$r = rand(0, 1)$
if $r < CR$ *then*
$Ui[j] = Vi[j]$
else
$r = rand(0, 1)$
if $r <= 0.7$ *then*
$Ui[j] =$ *Pbest*$[j]$
else
$Ui[j] =$ *GeneratedFromEquation(5)*
//CalculateFitness
fitness$P =$ *EvaluateFitness*(Pi)
fitness$U =$ *EvaluateFitness*(Ui)
//UpdateIndividualsToEnterTheNextGenerationPopulation
if fitness$U <$ *fitness*P *then*
$Pi = Ui$
//OutputTheOptimalIndividualAsTheInitialClusterCentreForK-Means
bestIndividual $=$ *BestIndividual*(P)
//PerformK-MeansClustering
clusters $=$ *KMeansClustering(bestIndividual)*
//OutputClusteringResults
return clusters
end function

The flowchart of the K-means clustering algorithm based on the improved differential evolution is shown in Figure 1.

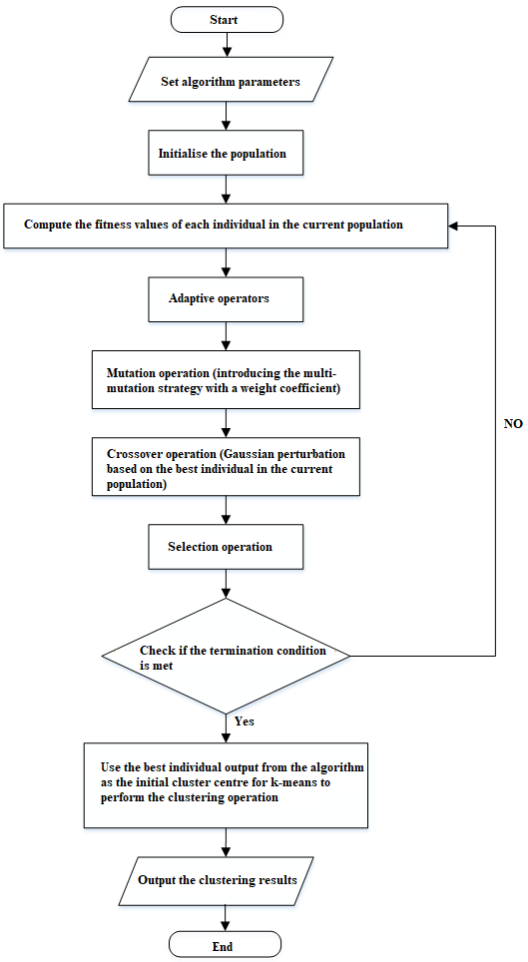

Figure 1. The flowchart of the K-means clustering algorithm based on the improved differential evolution

3 Experimental Results and Comparative Analysis

To validate the effectiveness of the proposed algorithm, extensive experiments were conducted on commonly used clustering datasets from the UCI public database, including Vowel, Iris, and Glass, as well as a synthetic dataset, Jcdx. Details of the datasets are provided in Table 1. The results of the proposed algorithm were compared with the traditional K-means clustering algorithm and the following enhanced K-means clustering algorithms: K-means based on discrete particle swarm optimisation (DPSO-KM), K-means based on grey wolf optimisation (GWO-KM), K-means based on whale optimisation (WOA-KM), and K-means based on differential evolution (DE-KM).

As for the experimental environment, the experiments were performed on a system running Windows 10, with an Intel Core i5-8300H processor and 16.0 GB of memory. Programming was conducted using PyCharm 2021.3.3.

Table 1. Details of experimental datasets

Dataset	Number of Samples	Feature Dimensions	Number of Clusters
Vowel	871	3	6
Iris	150	4	3
Glass	214	9	6
Jcdx	10,000	4	5

3.1 Evaluation Metrics

To assess the performance of the clustering algorithms, five evaluation metrics were selected: silhouette coefficient (SC), SSE, Davies-Bouldin (DB) index, Calinski-Harabasz (CH) index, and clustering time [10, 22].

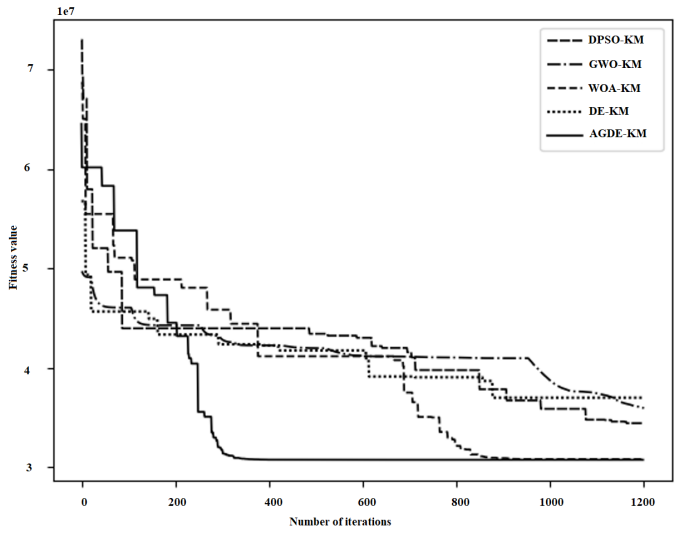

Figure 2. Comparison of the number of iterations for convergence on the Vowel dataset

a) SC

For a given sample x_i, let a_i denote the average distance between x_i and all other samples in the same cluster, and let $_{b_i}$ represent the minimum average distance between x_i and samples in other clusters. The SC for the sample is defined as follows:

$$s_i = \frac{b_i - a_i}{max\,(a_i, b_i)} \tag{7}$$

The SC for the clustering result is the mean value of s_i across all samples:

$$SC = \frac{1}{N}\sum_{i=1}^{N} s_i \tag{8}$$

A larger SC indicates better clustering performance.

b) SSE

$$SSE = \sum_{i,k} \left(x_{i,k} - c_k\right)^2 \tag{9}$$

where, $x_{i,k}$ represents a data point; and c_k denotes the centre of the cluster to which the data point belongs. A smaller SSE value signifies better clustering effectiveness.

c) DB index

$$DB = \frac{1}{k} \sum_{i=1}^{k} max_{j \neq i, i, j \in [1,K])} \frac{s_i + s_j}{d_{i,j}\left(c_i, c_j\right)} \tag{10}$$

where, s_i denotes the average distance between data points within cluster c_i and its centre; and $d_{i,j}$ represents the average distance between clusters c_i and c_j. A smaller DB value indicates better clustering effectiveness.

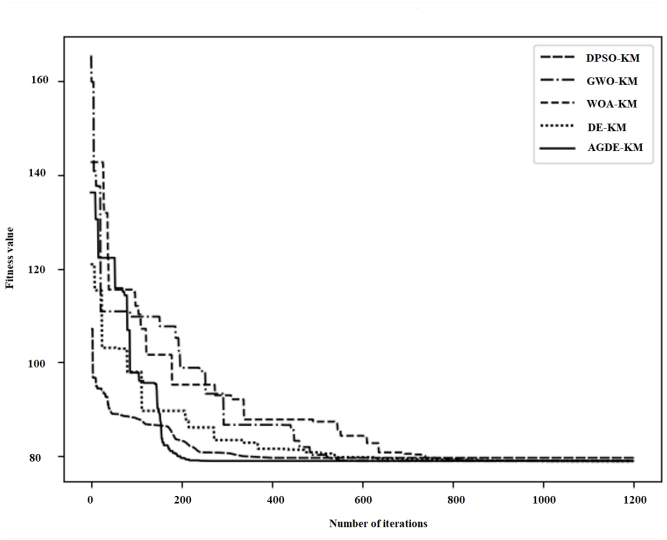

Figure 3. Comparison of the number of iterations for convergence on the Iris dataset

d) CH index

$$CH = \frac{tr\left(B_K\right)\left(K-1\right)}{tr\left(W_K\right)\left(N-K\right)} \tag{11}$$

where, B_K is the between-cluster covariance matrix; and W_K is the within-cluster covariance matrix.

$$W_k = \sum_{k=1}^{K} \sum_{x \in C_k} \left(x - c_k\right)\left(x - c_k\right)^T \tag{12}$$

$$B_K = \sum_{k=1}^{K} n_k \left(c_k - c_x\right)\left(c_k - c_x\right)^T \tag{13}$$

where, K represents the number of clusters; c_k represents a cluster set centered around c_k; n_k denotes the number of samples in set c_k; c_x is the centre of the dataset; and tr refers to the trace of the matrix. A larger CH value indicates better clustering effectiveness.

e) Clustering time

Clustering time is recorded as the total runtime of the algorithm under identical environmental conditions, measured in seconds.

3.2 Experimental Results and Analysis

Experiments were conducted on three public datasets and one synthetic dataset to evaluate the performance of the K-means, DPSO-KM, GWO-KM, WOA-KM, DE-KM, and the proposed AGDE-KM algorithms.

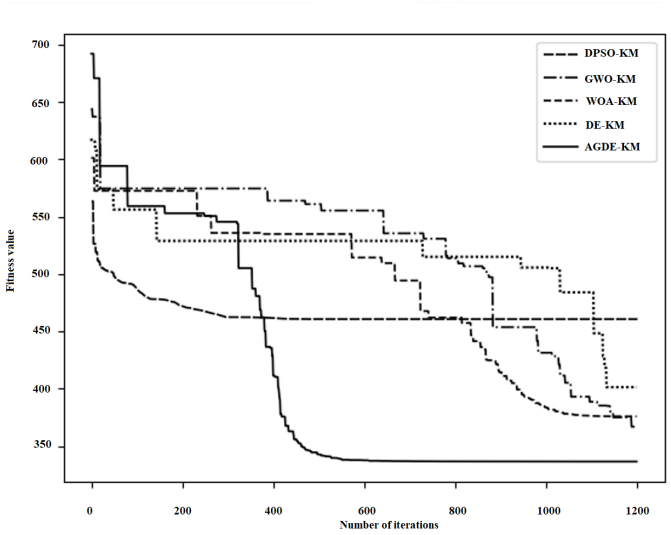

Figure 4. Comparison of the number of iterations for convergence on the Glass dataset

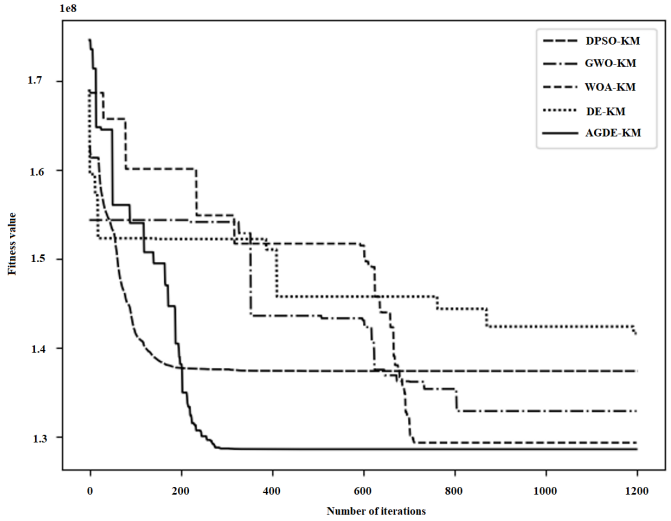

Figure 5. Comparison of the number of iterations for convergence on the Jcdx dataset

The parameter configurations for each algorithm are summarised in Table 2. Each algorithm was executed independently 30 times, and the experimental results are presented in Table 3, Table 4, Table 5, and Table 6. The convergence curves for the algorithms are shown in Figure 2, Figure 3, Figure 4, and Figure 5.

Table 2. Algorithm parameter configurations

Algorithm	Parameter Settings
DPSO-KM	Population size: 40; maximum velocity: 2.2; inertia weight: 0.9; self-learning factor: 0.6; social learning factor: 0.6; and maximum iterations: 1200
GWO-KM	Population size: 40; convergence factor (a) linearly decreases from 2 to 0; collaboration coefficient: random in $[0, 2]$; and maximum iterations: 1200
WOA-KM	Population size: 40; convergence factor (a) linearly decreases from 2 to 0; logarithmic spiral shape constant (b): 1; and maximum iterations: 1200
DE-KM	Population size: 40; mutation factor: 0.6; crossover factor: 0.5; and maximum iterations: 1200
AGDE-KM	Population size: ten times the solution dimensionality; mutation and crossover factors: adaptive values taken as per Eqs. (1) and (2); and maximum iterations: 1200

3.2.1 Comparison of clustering results

Table 3. Experimental results on the Vowel dataset

Algorithm	SC	SSE	DB	CH	Time (s)
K-means	0.3572	32534878.2374	0.9867	1374.9278	92.26
DPSO-KM	0.3620	31244793.4804	0.9526	1439.1136	36.45
GWO-KM	0.3617	31462481.2905	0.9542	1432.9681	41.75
WOA-KM	0.3625	30868331.4917	0.9463	1453.2708	19.27
DE-KM	0.3604	31719273.0195	0.9630	1414.7935	60.40
AGDE-KM	0.3628	30697028.0907	0.9438	1462.3049	15.66

Table 4. Experimental results on the Iris dataset

Algorithm	SC	SSE	DB	CH	Time (s)
K-means	0.5525	98.1774	0.7483	475.1548	5.57
DPSO-KM	0.5525	98.1774	0.7483	475.1548	5.57
GWO-KM	0.5611	78.9416	0.6631	560.3952	1.29
WOA-KM	0.5614	78.9408	0.6623	560.3999	1.12
DE-KM	0.5608	78.9424	0.6639	560.3863	1.82
AGDE-KM	0.5614	78.9408	0.6623	560.3999	1.04

Table 5. Experimental results on the Glass dataset

Algorithm	SC	SSE	DB	CH	Time (s)
K-means	0.3733	390.7793	1.0455	103.8238	17.13
DPSO-KM	0.4479	356.8504	0.9405	117.1880	9.03
GWO-KM	0.4625	342.9133	0.9186	122.3806	4.19
WOA-KM	0.4617	343.6558	0.9214	121.9510	5.72
DE-KM	0.4594	345.8267	0.9255	121.1542	7.18
AGDE-KM	0.4664	338.7536	0.9133	123.3459	3.86

Table 6. Experimental results on the Jcdx dataset

Algorithm	SC	SSE	DB	CH	Time (s)
K-means	0.3331	136627794.9409	1.1792	2444.7159	6774.61
DPSO-KM	0.3586	131276299.9431	1.1267	2647.7584	2743.30
GWO-KM	0.3629	130372315.6920	1.1179	2678.9136	2064.72
WOA-KM	0.3691	129074131.5716	1.1051	2728.8435	1084.51
DE-KM	0.3523	132598434.0014	1.1397	2595.1784	3739.62
AGDE-KM	0.3728	128294153.1227	1.0975	2756.6928	499.15

The comparisons of clustering results presented in Table 3 to Table 6 reveal that on the Iris dataset, due to the relatively uniform distribution of sample data and the small number of clusters, the improvement in clustering performance achieved by AGDE-KM over other algorithms is not substantial. However, the efficiency of clustering has been significantly enhanced. When the results across the Vowel, Iris, Glass, and Jcdx datasets are considered, AGDE-KM demonstrates clear advantages over the traditional K-means clustering algorithm. In terms of SC, the improvements are 1.57%, 1.61%, 24.94%, and 11.92%, respectively. In terms of SSE, the reductions are 5.65%, 19.59%, 13.31%, and 6.1%, respectively. In terms of DB index, the reductions are 4.35%, 11.49%, 12.64%, and 6.93%, respectively. In terms of CH index, the increases are 6.36%, 17.94%, 18.8%, and 12.76%, respectively. In terms of clustering time, the reductions are 83.03%, 81.33%, 77.47%, and 92.63%, respectively. From a theoretical perspective, the improvements can be attributed to the use of adaptive operators and the multi-mutation strategy in AGDE-KM. During the early stages of evolution, a larger search space is ensured, while in the later stages, the efficiency of local search and convergence speed is gradually accelerated as the number of iterations increases.

Furthermore, the Gaussian perturbation crossover operation, guided by the best individual in the current population, reduces the likelihood of the algorithm becoming trapped in local optima, particularly in datasets with numerous features and clusters. This enhancement improves the precision of the algorithm's optimisation capability. The results confirm that AGDE-KM exhibits superior clustering performance and greater applicability across diverse data environments.

3.2.2 Comparison of algorithm convergence

The convergence performance of the algorithms was validated using the curves shown in Figure 2 to Figure 5. The horizontal axis represents the number of iterations, while the vertical axis denotes the fitness value of the optimal solution found by the algorithm. The results demonstrate that, under the same number of iterations, AGDE-KM exhibits significantly faster convergence speeds and higher optimisation precision on the Vowel, Glass, and Jcdx datasets compared to the other four algorithms. On the Iris dataset, due to its relatively uniform sample distribution, the differences in optimisation precision are minimal; however, AGDE-KM still achieves notably faster convergence than the other algorithms. From a theoretical perspective, the improved algorithm, which incorporates adaptive operators and a multi-mutation strategy, retains a broad search space during the early stages of evolution. As the number of iterations increases, the effects of the improved operators and mutation strategies become more pronounced, progressively accelerating the convergence process. Additionally, the introduction of Gaussian perturbation crossover operation, guided by the best individual in the current population, reduces the likelihood of the algorithm becoming trapped in local optima. This enhancement further improves the optimisation precision of the algorithm.

4 Conclusion

To address the issues of poor stability and low efficiency in traditional K-means clustering caused by the random selection of initial cluster centres, AGDE-KM was proposed. AGDE-KM employs adaptive operators, incorporates a multi-mutation strategy with a weighted coefficient during the mutation phase, and balances the algorithm's global and local search capabilities, thereby accelerating convergence. To mitigate the risk of becoming trapped in local optima, a Gaussian perturbation crossover operation guided by the best individual in the current population was introduced. This operation provides superior evolutionary directions for individuals while preserving diversity across dimensions. This enhancement improves local search capabilities and avoids the problem of local optima. The optimal solution generated by the improved differential evolution algorithm was utilised as the initial cluster centre, replacing the initial cluster centre randomly selected by the traditional K-means. This approach effectively resolves the issues of poor clustering stability, suboptimal clustering effectiveness, and inefficiency. The improvements achieved make the clustering process more stable and better equipped to capture the intrinsic structure of data, thereby enhancing clustering quality and efficiency. The proposed algorithm provides an effective solution for clustering problems in diverse data environments, demonstrating robustness and high applicability.

Funding

This paper was supported by Baoding Science and Technology Research and Development Guidance Plan Project (Grant No.: 2311ZG011).

Data Availability

The data used to support the research findings are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] K. P. Sinaga and M. S. Yang, "Unsupervised K-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80 716–80 727, 2020. https://doi.org/10.1109/ACCESS.2020.2988796

[2] L. Sun, J. Zhang, W. Ding, and J. Xu, "Feature reduction for imbalanced data classification using similarity-based feature clustering with adaptive weighted k-nearest neighbors," *Inf. Sci.*, vol. 593, pp. 591–613, 2022. https://doi.org/10.1016/j.ins.2022.02.004

[3] J. Kacprzyk and W. Pedrycz, *Springer Handbook of Computational Intelligence*. Springer, 2015. https://doi.org/10.1007/978-3-662-43505-2

[4] I. K. Khan, H. B. Daud, N. B. Zainuddin, R. Sokkalingam, M. Farooq, M. E. Baig, G. Ayub, and M. Zafar, "Determining the optimal number of clusters by Enhanced Gap Statistic in K-mean algorithm," *Egypt. Inform. J.*, vol. 27, p. 100504, 2024. https://doi.org/10.1016/j.eij.2024.100504

[5] A. Ahmad and S. S. Khan, "initKmix-A novel initial partition generation algorithm for clustering mixed data using k-means-based clustering," *Expert Syst. Appl.*, vol. 167, p. 114149, 2021. https://doi.org/10.1016/j.eswa.2020.114149

[6] W. Guo, W. Wang, S. Zhao, Y. Niu, Z. Zhang, and X. Liu, "Density peak clustering with connectivity estimation," *Knowl.-Based Syst.*, vol. 243, p. 108501, 2022. https://doi.org/10.1016/j.knosys.2022.108501

[7] W. Lu, "Improved K-means clustering algorithm for big data mining under Hadoop parallel framework," *J. Grid Comput.*, vol. 18, no. 2, pp. 239–250, 2020. https://doi.org/10.1007/s10723-019-09503-0

[8] A. Dubey and A. Choubey, "A systematic review on K-means clustering techniques," *Int. J. Sci. Res. Eng. Technol.*, vol. 6, no. 6, 2017.

[9] Z. Zhang and J. Lan, "K-means clustering algorithm based on bee colony strategy," *J. Phys.: Conf. Ser.*, vol. 2031, no. 1, p. 012058, 2021. https://doi.org/10.1088/1742-6596/2031/1/012058

[10] M. Ay, L. Özbakır, S. Kulluk, B. Gülmez, G. Öztürk, and S. Özer, "FC-Kmeans: Fixed-centered K-means algorithm," *Expert Syst. Appl.*, vol. 211, p. 118656, 2023. https://doi.org/10.1016/j.eswa.2022.118656

[11] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," *Inf. Sci.*, vol. 622, pp. 178–210, 2023. https://doi.org/10.1016/j.ins.2022.11.139

[12] T. K. Biswas, K. Giri, and S. Roy, "ECKM: An improved K-means clustering based on computational geometry," *Expert Syst. Appl.*, vol. 212, p. 118862, 2023. https://doi.org/10.1016/j.eswa.2022.118862

[13] L. Bai, C. Song, L. Xu, Z. Song, and J. Jiang, "K-means clustering algorithm based on improved quantum revolving door artificial fish schooling algorithm and its applications," *Appl. Res. Comput.*, vol. 39, no. 3, pp. 797–801+806, 2022. https://doi.org/10.19734/j.issn.1001-3695.2021.08.0354

[14] L. Sun, Y. Zhang, C. Zhang, and J. Xu, "Optimization algorithm based on improved particle swarm and K-means clustering," *J. Jiangsu Univ. Sci. Technol. (Nat. Sci. Ed.)*, vol. 37, no. 3, pp. 81–90, 2023. https://doi.org/10.20061/j.issn.1673-4807.2023.03.013

[15] R. Wang, X. Cui, and Y. Li, "Self-adaptive adjustment of cuckoo search K-means clustering algorithm," *Appl. Res. Comput.*, vol. 35, no. 12, pp. 3593–3597, 2018. https://doi.org/10.3969/j.issn.1001-3695.2018.12.016

[16] S. Wang, J. Zhang, C. Chen, H. Zhang, and C. Ma, "Optimization of K-means based on variance statistics and improved swarm intelligent algorithm," *J. Syst. Sci. Math. Sci.*, vol. 38, no. 10, pp. 1117–1127, 2018. https://doi.org/10.12341/jssms13460

[17] X. Chen, Y. Wei, M. Ren, and Y. Meng, "Weighted K-means clustering algorithm based on firefly algorithm," *Appl. Res. Comput.*, vol. 35, no. 2, pp. 466–470, 2018. https://doi.org/10.3969/j.issn.1001-3695.2018.02.031

[18] X. Hu, L. Wang, H. Zhang, Y. Chang, and Y. Wang, "Research on improved K-means clustering algorithm based on lion group optimization," *Control Eng. China*, vol. 29, no. 11, pp. 1996–2002, 2022. https://doi.org/10.14107/j.cnki.kzgc.20220162

[19] Y. Huang, X. Qian, and W. Song, "Improving dual-population differential evolution based on hierarchical mutation and selection strategy," *Electron.*, vol. 13, no. 1, p. 62, 2023. https://doi.org/10.3390/electronics13010062

[20] Q. Yang, L. Cai, and Y. Xun, "A review of differential evolutionary algorithms," *Pattern Recognit. Artif. Intell.*, vol. 21, no. 4, pp. 506–513, 2008.

[21] F. Moodi and H. Saadatfar, "An improved K-means algorithm for big data," *IET Softw.*, vol. 16, no. 1, pp. 48–59, 2022. https://doi.org/10.1049/sfw2.12032

[22] Y. Wang, Y. Xiao, P. Zuo, B. Yang, Y. Liu, and Z. Duan, "Identification of traffic accident-prone points based on improved clustering algorithm," *Appl. Res. Comput.*, vol. 40, no. 10, pp. 2993–2999, 2023. https://doi.org/10.19734/j.issn.1001-3695.2023.02.0086