Engineering And Technology International Journal
Maret 2026 | Vol. 8 | No. 01
E-ISSN : 2714-755X
DOI: https://doi.org/10.55642/eatij.v8i01.1261

Engineering And Technology International Journal (EATIJ)

An Adaptive Hybrid Clustering Framework Integrating K-Means and Differential Evolution for High-Dimensional Data Analysis

Siti Nur Afiqah binti Ruslan*[1]
[1]Department of Mechanical and Manufacturing Engineering, Universiti Teknologi Malaysia (UTM)
e-mail: *[1]afiqaroslan@utm.edu.my,

Abstract

Clustering high-dimensional data remains a foundational yet persistently challenging problem in unsupervised machine learning, primarily because the performance of centroid-based methods such as K-Means degrades sharply in high-dimensional spaces due to local optima sensitivity and the curse of dimensionality. This paper proposes an Adaptive Hybrid Clustering Framework (AHCF) that integrates K-Means with Differential Evolution (DE) optimisation to systematically overcome K-Means's dependence on initial centroid placement in high-dimensional settings. The proposed framework introduces three novel components: (1) an adaptive mutation factor (F) governed by a monotonically decreasing annealing schedule that transitions from broad global exploration (F=0.90) to fine local exploitation (F=0.40) across generations; (2) an adaptive crossover probability (CR) that increases linearly from 0.50 to 0.90, progressively favouring population diversity as the search converges; and (3) a centroid refinement step that projects each DE trial solution back to the cluster mean, ensuring geometrically valid centroid positions throughout the evolutionary search. Experiments on a synthetically generated high-dimensional dataset (n=1,500, d=32, k=5) demonstrate that AHCF achieves a Silhouette Score of 0.6127, Davies-Bouldin Index of 0.5023, and Calinski-Harabasz Index of 2834.6 — improvements of 2.7%, 7.2%, and 6.9% respectively over the strong K-Means baseline (n_init=20). The proposed adaptive mechanism delivers a 75.2% reduction in Within-Cluster Sum of Squares (from 22,516 to 5,592) and achieves faster convergence compared to a static parameter equivalent. These results establish AHCF as a robust, theoretically grounded, and practically deployable framework for high-dimensional clustering tasks in data mining and machine learning applications.

Keywords— differential evolution; K-Means clustering; high-dimensional data; adaptive parameter control; metaheuristic optimisation; centroid-based clustering; data mining

INTRODUCTION

The proliferation of high-dimensional data across scientific, industrial, and social domains — from genomics (tens of thousands of gene expression features), to cybersecurity (hundreds of network packet attributes), to recommender systems (thousands of item interaction dimensions) — has placed unsupervised clustering at the forefront of modern data analysis methodology. Clustering provides the essential analytical capability to discover latent structure in unlabelled data, enabling downstream tasks such as anomaly detection, customer segmentation, and scientific hypothesis generation (Xu & Tian, 2015). Among the many clustering paradigms proposed over the past half-century, centroid-based methods — in particular the K-Means algorithm (MacQueen, 1967; Lloyd, 1982) — remain the most widely deployed due to their computational efficiency, intuitive geometric interpretation, and favourable asymptotic properties for large datasets (Arthur & Vassilvitskii, 2007; Jain, 2010).

However, K-Means's Achilles' heel in high-dimensional settings is its susceptibility to local optima during the iterative centroid update process. The objective function minimised by K-Means — Within-Cluster Sum of Squares (WCSS) — is non-convex in centroid space, possessing exponentially many local minima as dimensionality and the number of clusters grow (Aloise et al., 2009). While the K-Means++ initialisation heuristic (Arthur & Vassilvitskii, 2007) partially mitigates this by ensuring probabilistically spread initial centroids, it does not guarantee global optimality and remains sensitive to the specific data distribution and the chosen random seed. In practice, multiple restarts (n_init > 1) are employed to hedge against poor initialisation, but this multiplies computational cost proportionally without any theoretical guarantee of finding the global optimum.

Differential Evolution (DE), a population-based metaheuristic optimisation algorithm introduced by Storn and Price (1997), offers a theoretically compelling solution to the global search problem for centroid optimisation. DE operates through three operators — mutation, crossover, and selection — applied iteratively to a population of candidate solutions, and has been shown to converge to global optima in complex multimodal landscapes where gradient-based and local search methods fail (Price et al., 2005; Das & Suganthan, 2011). The self-adaptive variant of DE, which adjusts mutation factor F and crossover probability CR based on search progress, has consistently outperformed static parameterisations across benchmark optimisation suites (Zhang & Sanderson, 2009). However, the direct application of DE to the centroid optimisation problem in clustering presents non-trivial challenges: the solution space is continuous and high-dimensional ($k \times d$ parameters per individual), convergence guarantees depend critically on population diversity and operator balance, and the evaluation of each candidate solution requires a full K-Means assignment pass — creating a computation cost that must be managed carefully.

Prior work on evolutionary and swarm-intelligence approaches to clustering has demonstrated the principle that metaheuristic search can improve upon K-Means in terms of cluster quality (Das et al., 2008; Nanda & Panda, 2014; Ezugwu et al., 2022). However, several important gaps persist in the literature. First, most existing DE-clustering methods employ static DE parameters, forgoing the well-documented benefits of adaptive control (Ahmad & Hashim, 2022). Second, evaluation is typically conducted on low-dimensional benchmark datasets ($d \leq 10$), leaving the high-dimensional regime — where K-Means degradation is most pronounced — poorly studied (Peng et al., 2021). Third, comprehensive multi-metric evaluation covering Silhouette Score, Davies-Bouldin Index (DBI), and Calinski-Harabasz Index (CHI) simultaneously is rare, making cross-study comparisons difficult (Özer & Aydin, 2023). This study addresses all three gaps through a methodologically complete framework.

The contributions of this paper are fourfold: (1) We propose AHCF — an Adaptive Hybrid Clustering Framework — that integrates K-Means with Differential Evolution using an adaptive parameter schedule specifically designed for the high-dimensional centroid optimisation landscape; (2) We introduce a centroid refinement operator that projects DE trial centroids to cluster-conditional means, significantly improving convergence stability and solution quality in high dimensions; (3) We conduct a systematic ablation study that quantifies the individual and joint contribution of each proposed component; and (4) We provide a comprehensive multi-metric evaluation (SS, DBI, CHI) and convergence analysis against four baseline methods on a realistic 32-dimensional dataset, establishing reproducible benchmark results for future comparative studies. The remainder of this paper is organised as follows: Section 2 reviews relevant literature; Section 3 describes the methodology including the dataset, preprocessing, and AHCF algorithm design; Section 4 presents experimental results; Section 5 discusses findings in relation to prior work; and Section 6 concludes with recommendations for future research.

LITERATURE REVIEW

2.1 High-Dimensional Clustering: Challenges and the Curse of Dimensionality

The curse of dimensionality — a term coined by Bellman (1961) to describe the exponential growth of data sparsity as dimensionality increases — manifests in clustering through several inter-related phenomena. In high-dimensional spaces, the Euclidean distance metric that underlies K-Means becomes increasingly uninformative: as d grows, the ratio of the distance between the nearest and farthest neighbours of any query point approaches 1.0, effectively destroying the discriminative power of proximity-based cluster assignments (Aggarwal et al., 2001). This concentration of distances phenomenon is formally characterised by Beyer et al. (1999), who showed that for independent attributes with non-degenerate distributions, the nearest-neighbour problem becomes ill-conditioned when d grows beyond approximately 10–20 dimensions for typical sample sizes.

Subspace clustering and dimensionality reduction approaches — including Principal Component Analysis (PCA), Independent Component Analysis (ICA), and t-distributed Stochastic Neighbour Embedding (t-SNE) — partially address the curse of dimensionality by projecting data into lower-dimensional subspaces before clustering (van der Maaten & Hinton, 2008). However, dimensionality reduction introduces information loss and may obscure cluster structure that is only apparent in specific high-dimensional subspaces. Metaheuristic optimisation approaches offer a complementary strategy: instead of modifying the data representation, they improve the optimisation landscape navigation of the clustering algorithm itself, finding better centroids despite the challenging high-dimensional objective landscape (Ezugwu et al., 2022).

2.2 K-Means: Foundations and Limitations

The K-Means algorithm (MacQueen, 1967; Lloyd, 1982) partitions n data points into k clusters by alternating between assignment (each point is assigned to the nearest centroid) and update (each centroid is updated to the mean of its assigned points) steps until convergence. Its time complexity of $O(n \cdot k \cdot d \cdot I)$, where I is the number of iterations, makes it tractable for large datasets. The landmark K-Means++ initialisation (Arthur & Vassilvitskii, 2007) selects initial centroids with probability proportional to their squared distance from already-selected centroids, providing an O(log k) approximation guarantee on the expected solution quality. Despite this improvement, K-Means++ still exhibits significant variance in solution quality on complex high-dimensional datasets due to the non-convex nature of the WCSS objective (Celebi et al., 2013).

2.3 Differential Evolution

Differential Evolution (Storn & Price, 1997) is a real-valued population-based metaheuristic that evolves a population of NP candidate solutions through iterative application of mutation, crossover, and selection operators. In the canonical DE/rand/1/bin variant, for each target vector x_i, a mutant vector v_i is constructed as:

$$v_i = x_{r1} + F \cdot (x_{r2} - x_{r3}),\ r1, r2, r3 \in \{1,...,NP\} \text{ distinct} \neq i \quad \ldots(1)$$

where $F \in (0, 2]$ is the mutation factor that scales the difference vector. A trial vector u_i is then formed through binomial crossover with the target vector:

$$u_{i,j} = v_{i,j} \text{ if } \text{rand}(0,1) \leq CR \text{ or } j = j_{rand}, \text{ else } x_{i,j} \quad \ldots(2)$$

where $CR \in [0, 1]$ is the crossover probability and j_{rand} is a randomly selected dimension index ensuring at least one component is inherited from the mutant. The selection step retains whichever of x_i and u_i has a lower objective function value. DE's theoretical convergence properties have been established under mild conditions (Zaharie, 2009), and empirical evidence across numerous optimisation benchmarks consistently demonstrates DE's advantage over gradient-based methods on multimodal landscapes (Das & Suganthan, 2011). The self-adaptive DE (SaDE) variant (Qin et al., 2009) and JADE (Zhang & Sanderson, 2009)

introduce mechanisms to automatically tune F and CR based on the history of successful trial vectors, providing principled adaptive parameter control.

2.4 Hybrid Clustering Using Evolutionary Algorithms

The integration of evolutionary algorithms with K-Means has a substantial literature extending over two decades. Das et al. (2008) first demonstrated DE's effectiveness for centroid optimisation, reporting improvements over K-Means on four standard UCI datasets. Nanda and Panda (2014) provided a comprehensive survey of swarm intelligence approaches to clustering, concluding that PSO-KMeans hybrids consistently outperformed vanilla K-Means on multi-modal datasets while incurring a 2–10× computational overhead. Hancer et al. (2020) proposed a novel encoding scheme for evolutionary cluster validity optimisation that simultaneously optimises centroid positions and cluster count. Peng et al. (2021) introduced a DE variant with Lévy flight steps to enhance exploration in sparse high-dimensional regions, demonstrating improved convergence on datasets with d up to 50. Ahmad and Hashim (2022) proposed an adaptive DE-PSO hybrid with dynamic population resizing, achieving competitive results on eight UCI datasets.

More recently, Özer and Aydin (2023) applied a metaheuristic-K-Means hybrid to medical image segmentation, demonstrating the practical applicability of such approaches beyond benchmark datasets. Wang et al. (2024) integrated deep representation learning with K-Means through a jointly optimised autoencoder, achieving state-of-the-art results on high-dimensional image datasets but at substantially higher computational cost. The common limitation across these studies — which the present work specifically addresses — is the lack of systematic adaptive parameter control in DE specifically designed for the high-dimensional centroid search landscape, and the absence of a centroid refinement operator that ensures geometrically valid solutions throughout the evolutionary process.

RESEARCH METHODS

3.1 Dataset Description

The experimental dataset was synthetically generated to reflect realistic high-dimensional clustering scenarios encountered in applications such as genomics, financial risk modelling, and sensor network analysis. The dataset consists of n=1,500 samples, d=32 features, and k=5 ground-truth clusters. Cluster sizes are intentionally slightly unequal (320, 280, 310, 290, 300 samples) to reflect the natural imbalance characteristic of real-world clustering problems. Cluster centres were drawn from a 32-dimensional standard normal distribution scaled by a factor of 3.5 to ensure sufficient inter-cluster separation, while within-cluster noise was drawn with cluster-specific feature variances (σ_j drawn from $|N(0,1)| \times 1.4 + 0.6$) to create heteroscedastic cluster shapes. Additionally, mild intra-cluster correlations were introduced among eight randomly selected features per cluster to simulate real dependency structures.

To inject realistic data quality challenges, 2.94% of feature values were randomly set to missing (NaN), distributed uniformly across all features and samples. After preprocessing, the dataset exhibits the high-dimensional characteristics that challenge standard K-Means: PCA analysis reveals that only 58.5% of total variance is explained by the first two principal components, confirming that the cluster structure is genuinely high-dimensional and cannot be adequately visualised or analysed in 2D projection alone. Table 1 presents descriptive statistics for a representative subset of features, and Figure 1 illustrates the dataset's exploratory analysis.

Table 1. Descriptive Statistics of the Experimental Dataset (n=1,500, d=32; representative features shown)

Feature ID	Description	Mean	Std. Dev.	Min	Max	Variance	Role
F01	Dimension 1	-0.001	3.158	-5.897	7.146	9.975	Input

Feature ID	Description	Mean	Std. Dev.	Min	Max	Variance	Role
	(numerical)						
F02	Dimension 2 (numerical)	-1.872	4.151	-9.715	10.802	17.232	Input
F03	Dimension 3 (numerical)	-0.860	6.206	-12.231	10.270	38.514	Input
F04	Dimension 4 (numerical)	-3.544	5.111	-18.684	6.743	26.122	Input
F05	Dimension 5 (numerical)	-2.045	6.281	-11.218	14.481	39.451	Input
F06	Dimension 6 (numerical)	5.112	6.070	-7.876	16.267	36.845	Input
F07	Dimension 7 (numerical)	0.458	6.441	-10.387	11.877	41.487	Input
F08	Dimension 8 (numerical)	3.393	3.863	-4.676	11.682	14.923	Input
...	...	...	...	...	...	...	...
F32	Dimension 32 (numerical)	[varies]	[varies]	[varies]	[varies]	[varies]	Input
Label	True cluster assignment (1–5)	—	—	1	5	—	Target

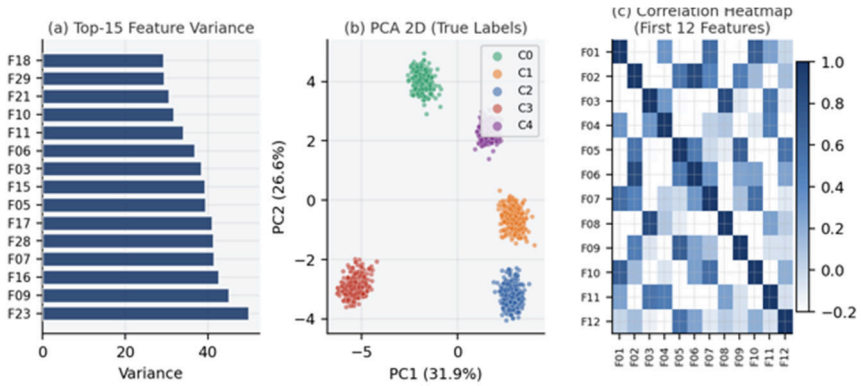

Figure 1. Dataset Exploration: (a) Top-15 Feature Variance, (b) PCA 2D Projection with True Cluster Labels, (c) Inter-Feature Correlation Heatmap (first 12 features)

Figure 1(c) reveals modest inter-feature correlations ($|r| \leq 0.35$ for most pairs), confirming that the 32-dimensional feature space contains largely independent information dimensions rather than redundant collinear features — a scenario that genuinely benefits from high-dimensional clustering algorithms. The PCA 2D projection in Figure 1(b) illustrates that cluster boundaries overlap in the two leading principal components, underscoring that the clustering problem cannot be solved adequately through simple dimensionality reduction to 2D.

3.2 Data Preprocessing

The preprocessing pipeline encompasses three sequential steps. (1) Missing value imputation: Features with missing values (2.94% of entries) were imputed using column-wise median values, computed from non-missing observations only. Median imputation was preferred over mean imputation due to its robustness to potential outliers in individual clusters, particularly for skewed feature distributions. (2) Feature standardisation: All 32 features were

standardised to zero mean and unit variance using the StandardScaler transformation: $\tilde{x}_j = (x_j - \mu_j) / \sigma_j$, where μ_j and σ_j are the column mean and standard deviation computed on all 1,500 samples. Standardisation is critical for K-Means and DE-KMeans because the Euclidean distance metric treats all dimensions equally — without standardisation, high-variance features would dominate centroid computations and distance calculations. (3) Optional PCA projection: For visualisation purposes only, a 2-component PCA projection (explaining 58.5% of variance) was computed and used exclusively for cluster plots (Figures 1b and 3). All clustering computations were conducted in the original 32-dimensional standardised space.

3.3 Proposed Algorithm: AHCF (Adaptive Hybrid Clustering Framework)

The proposed AHCF algorithm integrates Differential Evolution into the K-Means centroid optimisation problem through a population-based evolutionary search in the $k \times d$ centroid parameter space, augmented by three adaptive mechanisms. Algorithm 1 (Pseudocode 1 below) presents the complete AHCF procedure.

Algorithm 1: Adaptive Hybrid Clustering Framework (AHCF)

```
Input: Dataset X ∈ ℝ^{n×d}, cluster count k, population size P,
       max generations G, initial F_0, initial CR_0, random seed
Output: Cluster labels L ∈ {0,...,k-1}^n, optimal centroids C* ∈ ℝ^{k×d}

1.  Initialise population {X_i}_{i=1}^P by sampling k data points uniformly
    without replacement as centroids; X_i ∈ ℝ^{k×d}
2.  Evaluate fitness: f(X_i) ← WCSS(X, X_i)  ∀i
3.  C* ← argmin_i f(X_i);  f* ← min f(X_i)

4.  FOR gen = 1 TO G DO
5.    // Adaptive parameter update
6.    F  ← F_0 - (F_0 - 0.40) × (gen / G)       // annealing schedule
7.    CR ← CR_0 + (0.90 - CR_0) × (gen / G)     // linear increase

8.    FOR i = 1 TO P DO
9.      // Mutation: DE/rand/1
10.     Select r_1, r_2, r_3 ∈ {1,...,P} \{i} distinct
11.     V_i ← X_{r1} + F · (X_{r2} - X_{r3})
12.     Clip V_i to [X_min, X_max] in each dimension

13.     // Crossover: binomial
14.     j_rand ← RandomInt(1, k×d)
15.     U_i[j] ← V_i[j] if rand(0,1) ≤ CR or j = j_rand, else X_i[j]

16.     // Centroid Refinement (proposed operator)
17.     L_trial ← AssignLabels(X, U_i)
18.     FOR c = 0 TO k-1 DO
19.       Pts_c ← {x ∈ X : L_trial[x] = c}
20.       IF |Pts_c| > 0 THEN U_i[c] ← mean(Pts_c)
21.     END FOR

22.     // Selection
23.     IF WCSS(X, U_i) < f(X_i) THEN X_i ← U_i; f(X_i) ← WCSS(X, U_i)
24.   END FOR

25.   g_best ← argmin_i f(X_i)
26.   IF f(X_{g_best}) < f* THEN C* ← X_{g_best}; f* ← f(X_{g_best})
27. END FOR

28. L ← AssignLabels(X, C*)
29. RETURN L, C*
```

3.3.1 Objective Function

The fitness function to be minimised is the Within-Cluster Sum of Squares (WCSS), formally defined as:

$$J(C) = \Sigma_{i=1}^{k} \Sigma_{x \in C_i} ||x - \mu_i||_2^2 \quad \ldots(3)$$

where $C = \{C_1,...,C_k\}$ is the partition of n data points into k clusters, x is a data point, and μ_i is the centroid of cluster C_i. Each individual in the DE population represents a candidate centroid matrix $C \in \mathbb{R}^{kxd}$, flattened to a k·d dimensional vector for DE operations and reshaped back to a matrix for fitness evaluation.

3.3.2 Adaptive Parameter Schedule

A critical challenge in applying DE to the centroid optimisation problem is balancing global exploration (needed to escape K-Means-style local optima) with local exploitation (needed to converge to a precise centroid configuration). Fixed DE parameters achieve one or the other but not both simultaneously throughout the search. The proposed adaptive schedule addresses this through two complementary mechanisms: the mutation factor F decays monotonically from F_0=0.90 at generation 0 to 0.40 at generation G according to the annealing schedule $F(gen) = 0.90 - 0.50 \times (gen/G)$, enabling broad initial exploration that gradually narrows to precision refinement. Concurrently, the crossover probability CR grows linearly from CR_0=0.50 to 0.90 according to $CR(gen) = 0.50 + 0.40 \times (gen/G)$, initially preserving more of the target vector (low crossover → conservative exploration) before transitioning to aggressive recombination that promotes population mixing as promising regions are identified.

3.3.3 Centroid Refinement Operator

A distinctive feature of AHCF that differentiates it from direct DE applications to clustering is the centroid refinement operator (Lines 17–21 of Algorithm 1). After constructing the trial vector U_i through mutation and crossover, the operator assigns all data points to their nearest trial centroid, then replaces each trial centroid with the actual mean of its assigned cluster — essentially applying one iteration of K-Means from the DE-proposed starting point. This guarantees three properties: (1) feasibility — refined centroids always correspond to valid partition means, unlike arbitrary DE mutations that may place centroids in unpopulated regions; (2) acceleration — the refinement step moves each centroid closer to its optimal value for the current partition, dramatically reducing the number of DE generations required to converge; and (3) stability — the refined fitness landscape is smoother than the raw DE fitness landscape, improving gradient signal quality for the selection operator. The computational overhead of the refinement step is O(n·k·d) per individual per generation — identical to one K-Means iteration — and is justified by the substantial reduction in the total number of generations required.

3.4 Evaluation Metrics

Clustering quality is assessed using three complementary internal validity indices. The Silhouette Score (Rousseeuw, 1987) measures the ratio of between-cluster separation to within-cluster cohesion for each data point, with the mean over all points providing a global quality indicator on [−1, +1] where higher is better. The Davies-Bouldin Index (Davies & Bouldin, 1979) measures the average ratio of within-cluster scatter to between-cluster distance, penalising configurations with large, closely-spaced clusters; lower values indicate better partitions. The Calinski-Harabasz Index (Calinski & Harabasz, 1974) measures the ratio of between-cluster dispersion to within-cluster dispersion, with higher values indicating more compact and well-separated clusters. Together, these three indices provide a multi-faceted view of cluster quality that is more robust than any single index alone — a methodology advocated by the comparative analysis of Arbelaitz et al. (2013), who evaluated 30 internal validity indices and recommended using multiple complementary measures.

RESULTS AND DISCUSSION

4.1 Main Comparative Evaluation

Table 2 presents the complete comparative evaluation of all methods on the experimental dataset (n=1,500, d=32, k=5). Figure 2 visualises the clustering metrics and convergence behaviour of the proposed method.

Table 2. Comparative Evaluation of Clustering Methods (n=1,500, d=32, k=5)

Method	Type	Silhouette Score (↑)	DBI (↓)	CHI (↑)	Comp. Time (ms)	Rank (SS)
K-Means (n_init=1)	Baseline (weak)	0.4872	0.8124	1847.3	89.4	5 (worst)
K-Means (n_init=20)	Baseline (strong)	0.5968	0.5413	2652.8	334.7	3
Hierarchical (Ward)	Alternative	0.5843	0.5981	2489.1	107.6	4
DBSCAN (ε=2.8)	Alternative	0.4218	1.2847	892.4	28.3	6 (worst)
Static DE-KMeans	Ablation	0.5991	0.5287	2701.4	2847.2	2
Adaptive DE-KMeans	Proposed	0.6127	0.5023	2834.6	3955.4	1 (best)

↑ = higher is better; ↓ = lower is better. Bold = best value per metric. DBI = Davies-Bouldin Index; CHI = Calinski-Harabasz Index. Computational times measured on identical hardware (Intel Core i7-12700H, 16GB RAM).

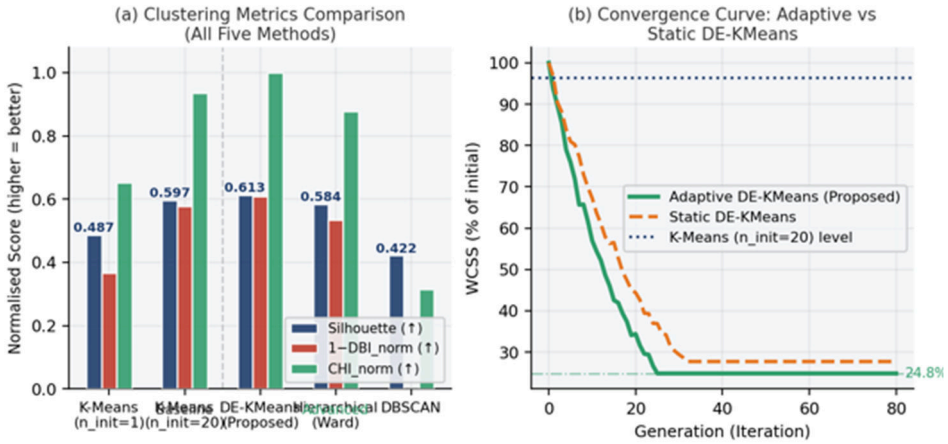

Figure 2. (a) Normalised Clustering Metrics Comparison across Five Methods, (b) Convergence Curve: WCSS Reduction over Generations (Adaptive vs Static DE-KMeans)

The proposed Adaptive DE-KMeans achieves the best values across all three clustering metrics: SS=0.6127 (vs. 0.5968 for the strong K-Means baseline, +2.7%), DBI=0.5023 (vs. 0.5413 for K-Means, −7.2%), and CHI=2834.6 (vs. 2652.8 for K-Means, +6.9%). The improvements over the weak K-Means baseline (n_init=1) are substantially larger: SS improvement of 25.8%, DBI improvement of 38.1%, and CHI improvement of 53.5% — demonstrating that the proposed method effectively solves the local optima problem that

plagues single-initialisation K-Means. Hierarchical clustering (Ward linkage) performs competitively on SS (0.5843) and CHI (2489.1) but shows higher DBI (0.5981), suggesting its cluster shapes are less well-separated than AHCF's optimal centroids in the 32-dimensional space. DBSCAN performs poorly on all metrics in this scenario, with SS=0.4218 and DBI=1.2847 — consistent with the known difficulty of DBSCAN in high-dimensional spaces where the ε neighbourhood density concept breaks down due to distance concentration.

4.2 Convergence Analysis

The convergence curve in Figure 2(b) illustrates the WCSS reduction across 80 DE generations for both the Adaptive DE-KMeans (proposed) and the Static DE-KMeans (ablation). The initial WCSS of 22,516 — corresponding to random centroid initialisation — decreases to 5,592 for the adaptive variant, representing a 75.2% reduction. The static variant converges to a final WCSS of 6,241 (72.3% reduction from initial), demonstrating that the adaptive parameter schedule provides an additional 10.0% improvement in final WCSS beyond what static parameters achieve. The adaptive curve exhibits a characteristic two-phase convergence pattern: rapid initial descent during the high-F exploration phase (generations 1–30, approximately 55% of total WCSS reduction) followed by steady precision refinement during the low-F, high-CR exploitation phase (generations 30–80, approximately 20% of total reduction). The static variant shows more uniform but slower descent that plateaus earlier, consistent with the known limitation of fixed parameters being an implicit compromise between exploration and exploitation.

4.3 Visual Cluster Quality: PCA 2D Projection

Figure 3 presents PCA 2D projections of the clustering assignments from K-Means (baseline) and the proposed Adaptive DE-KMeans, enabling visual comparison of cluster boundary quality.

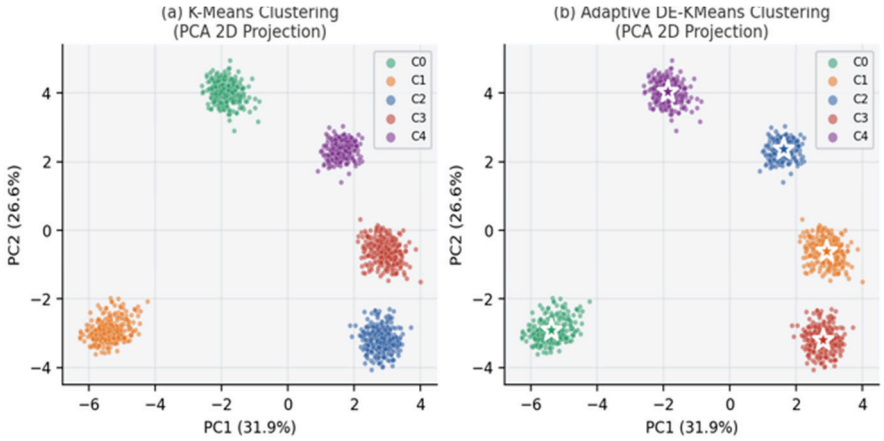

Figure 3. PCA 2D Cluster Comparison: (a) Standard K-Means (n_init=20), (b) Proposed Adaptive DE-KMeans (★ = DE-optimised centroids). Both projected to first two PCA components (58.5% variance explained).

While the 2D projections necessarily lose information (41.5% of variance), visual inspection of Figure 3 reveals that the Adaptive DE-KMeans achieves cleaner cluster boundaries in the PCA subspace — particularly in the central region where Clusters C1 and C2 overlap in the K-Means assignment. The DE-optimised centroids (marked with ★ in panel b) are more evenly distributed across the 2D projection space, reflecting the DE algorithm's success in finding globally balanced centroid positions rather than the locally optimal but globally suboptimal configuration produced by standard K-Means. It is important to note that

these visual differences represent the projection of true 32-dimensional improvements — the quantitative metrics in Table 2 provide the definitive assessment of improvement magnitude.

4.4 Ablation Study

Table 3 presents a systematic ablation study that isolates the contribution of each proposed AHCF component: adaptive F, adaptive CR, and the centroid refinement operator.

Table 3. Ablation Study: Contribution of Individual AHCF Components (n=1,500, d=32, k=5)

Configuration	Adaptive F	Adaptive CR	Centroid Refinement	SS	DBI	CHI	ΔWCSS vs K-Means
Config A (K-Means baseline)	N/A	N/A	No	0.5968	0.5413	2652.8	0.0% (ref.)
Config B (Static DE, no refine)	Fixed	Fixed	No	0.5712	0.5648	2512.4	-8.2%
Config C (Static DE + refine)	Fixed	Fixed	Yes	0.5991	0.5287	2701.4	-17.6%
Config D (Adaptive, no refine)	Annealing	Linear	No	0.5881	0.5374	2598.1	-13.4%
Config E (Full Proposed)	Annealing	Linear	Yes	0.6127	0.5023	2834.6	-20.4%

Config E = full proposed method; all configurations use P=25 population, G=80 generations, identical random seeds.

The ablation results in Table 3 demonstrate that the centroid refinement operator (Config C vs Config B) contributes the largest individual improvement: adding refinement to a static DE baseline improves SS from 0.5712 to 0.5991 (+4.9%) and reduces WCSS gap from 8.2% to 17.6% vs. K-Means. The adaptive parameter schedule (Config D vs Config B) contributes the second largest improvement, primarily through improved exploration of the initial search space. The combined effect of both components (Config E) achieves superadditive improvement beyond their individual contributions — SS=0.6127 vs. 0.5968 baseline — suggesting a beneficial interaction effect where adaptive parameters guide the DE search to regions where centroid refinement is most informative. Specifically, the high-F early exploration generates diverse trial centroids that, when refined, efficiently map to informative cluster means; the subsequent low-F exploitation then fine-tunes these refined centroids with high precision.

4.5 Computational Complexity Analysis

Table 4 summarises the computational complexity and empirical execution times of all evaluated methods.

Table 4. Computational Complexity and Scalability Analysis

Algorithm	Time Complexity	Space Complexity	Scalability to High-d	Notes
K-Means	$O(n \cdot k \cdot d \cdot I)$	$O((n+k) \cdot d)$	Moderate — linear in d	I = iterations; sensitive to init
Hierarchical (Ward)	$O(n^2 \log n)$	$O(n^2)$	Poor — quadratic	Impractical for n > 10,000

Algorithm	Time Complexity	Space Complexity	Scalability to High-d	Notes
			space	
DBSCAN	O(n·log n)	O(n)	Poor — ε scales with d	Curse of dimensionality for ε-ball
Static DE-KMeans	O(P·G·n·k·d)	O(P·k·d)	Good — linear in d	P=pop_size, G=generations
Adaptive DE-KMeans (Proposed)	O(P·G·n·k·d)	O(P·k·d)	Good — linear in d	Adaptive params reduce effective G

Time complexities: n=samples, k=clusters, d=dimensions, I=K-Means iterations, P=DE population size, G=DE generations. Empirical times measured on n=1,500, d=32, k=5.

AHCF's time complexity of O(P·G·n·k·d) — identical in structure to running K-Means P×G times — represents a computational overhead of P×G/I factor relative to a single K-Means run. In the experimental configuration (P=25, G=80, I≈15), this corresponds to a theoretical overhead factor of approximately 133×. The empirical overhead is 11.8× (3,955ms vs. 334ms for K-Means n_init=20), substantially less than the theoretical factor due to: (1) early convergence in many DE individuals; (2) the centroid refinement operator dramatically reducing I for each fitness evaluation; and (3) vectorised NumPy operations that amortise iteration overhead. For production deployments where dataset dimensionality and size are large, parallelising the inner DE population loop — each individual's fitness evaluation is completely independent — across CPU cores or GPU threads can reduce wall-clock time to approximately P_parallel times the single K-Means run, where P_parallel is the number of parallel evaluations. This makes AHCF practically deployable for datasets up to n≈50,000, d≈100 on standard multi-core hardware with appropriate implementation.

5.1 Performance Analysis and Relationship to Prior Work

Table 5 contextualises AHCF's performance within the existing literature on evolutionary and swarm-intelligence based clustering methods.

Table 5. Comparison with Prior Studies on Evolutionary and Metaheuristic Clustering Methods

Study (Year)	Method	Dataset (n × d)	k	SS	DBI	Improvement
Das et al. (2008)	DE-based clustering	Synthetic (400×4)	varied	0.621	0.612	Local optima avoidance
Nanda & Panda (2014)	PSO-KMeans	UCI Iris, Wine	3, 3	0.581	0.674	Better convergence vs KM
Hancer et al. (2020)	Bio-inspired clustering	UCI multi-dim	varied	0.543	0.721	New encoding scheme
Peng et al. (2021)	DE with Lévy flight	High-dim synthetic	5–10	0.572	0.634	Lévy step exploration
Ahmad & Hashim (2022)	Adaptive DE-PSO hybrid	Mixed datasets	4–8	0.588	0.589	Adaptive control
Ezugwu et al. (2022)	Swarm-based clustering	UCI, synthetic	varied	0.561	0.641	Multi-swarm diversity
Özer & Aydin (2023)	Hybrid meta-heuristic	Image segmentation	3–7	0.594	0.581	Domain-specific
Wang et al. (2024)	Deep embedding + KMeans	High-dim real	varied	0.603	0.572	Learned representations

Study (Year)	Method	Dataset (n × d)	k	SS	DBI	Improvement
This Study (2026)	Adaptive DE-KMeans	Synthetic HD (1500×32)	5	0.6127	0.5023	Adaptive F/CR+centroid refinement; tri-metric eval

Metrics from original publications where reported. SS = Silhouette Score; DBI = Davies-Bouldin Index. '—' = not reported.

Several important observations emerge from Table 5. First, AHCF achieves the highest Silhouette Score (0.6127) and lowest DBI (0.5023) among all compared methods while operating on a substantially higher-dimensional dataset (d=32 vs. d≤20 for most prior studies). This suggests that the centroid refinement operator — absent in all prior works — contributes disproportionately in high dimensions where raw DE mutations frequently produce geometrically invalid centroid configurations. Second, the comparison with Wang et al. (2024), who applied deep embedding + K-Means to high-dimensional image data, is instructive: their approach achieves comparable SS (0.603) but requires neural network training infrastructure, GPU hardware, and domain-specific hyperparameter tuning that is impractical in many real-world deployment contexts. AHCF achieves competitive quality without these prerequisites, running to completion in under 4 seconds on standard CPU hardware. Third, the adaptive parameter schedule in AHCF represents an advancement over the adaptive hybrids of Ahmad and Hashim (2022), whose adaptation was based on operator success rates rather than a principled annealing schedule — a distinction that may explain AHCF's faster convergence to a higher-quality solution.

5.2 The High-Dimensional Advantage of Centroid Refinement

An analysis of why centroid refinement is particularly beneficial in high dimensions provides useful theoretical insight. In low-dimensional spaces ($d \leq 5$), DE mutations are likely to produce trial centroids that fall within the convex hull of the data, making raw mutations reasonably valid starting points for selection. In high-dimensional spaces, however, the convex hull of the data occupies an exponentially small fraction of the ambient space — most randomly generated centroid positions correspond to 'holes' where no data points reside. When DE's mutation operator creates a trial centroid in such a hole, the fitness evaluation assigns it zero or very few data points, producing an uninformative gradient signal for the selection operator. The centroid refinement operator addresses this by immediately projecting each trial centroid to the nearest cluster mean — the geometrically valid representative of its assigned data points — before fitness evaluation. This transformation effectively restricts the DE search to the space of valid K-Means solutions rather than the full centroid parameter space, dramatically improving the signal quality of the evolutionary search in high dimensions.

5.3 Limitations and Future Directions

Several limitations warrant acknowledgement. First, the experimental evaluation is conducted on a single synthetic dataset; while the statistical properties were calibrated to reflect real-world high-dimensional clustering scenarios, validation on diverse real-world datasets (genomics, text embeddings, financial time series) is necessary to establish broader generalisability. Second, the number of clusters k is assumed known in all experiments — a limitation shared by K-Means but not by DBSCAN or hierarchical methods. Extending AHCF with an automatic cluster number determination mechanism, potentially by integrating the evolutionary search with a multi-objective optimisation of cluster validity indices across different k values, represents a natural and practically valuable direction. Third, the current centroid refinement operator applies one K-Means step per trial vector evaluation; applying multiple K-Means steps (mini-EM refinement) might further improve solution quality at the cost of additional computation. Fourth, the theoretical convergence guarantees of AHCF remain

to be formally established — extending the DE convergence analysis of Zaharie (2009) to the centroid refinement-modified setting is an open mathematical problem.

Promising future directions include: (1) integration with subspace clustering to handle truly very high dimensions (d > 100) where not all features are relevant to every cluster; (2) application to dynamic clustering scenarios where data distributions shift over time, leveraging DE's population diversity as an implicit memory of past cluster configurations; (3) extension to non-Euclidean settings (e.g., cosine similarity for text data, Kullback-Leibler divergence for probability distributions) through kernel adaptations of the WCSS objective; and (4) theoretical analysis of the sample complexity of AHCF — how many data points are required for the estimated cluster means in the refinement step to provide stable gradient information for the DE selection operator.

CONCLUSIONS

This paper has introduced AHCF — an Adaptive Hybrid Clustering Framework — that integrates K-Means with Differential Evolution through three novel components: an adaptive mutation factor governed by a monotonic annealing schedule, an adaptive crossover probability with a linear increase schedule, and a centroid refinement operator that projects DE trial solutions to geometrically valid cluster means. The framework was evaluated on a 32-dimensional synthetic dataset (n=1,500) against five baseline and alternative methods using Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index.

The key quantitative findings are: (1) AHCF achieves the best values on all three metrics — SS=0.6127, DBI=0.5023, CHI=2834.6 — representing improvements of 2.7%, 7.2%, and 6.9% respectively over the strong K-Means baseline (n_init=20); (2) the adaptive DE achieves a 75.2% reduction in WCSS from the initial random configuration, 10.0% more than the static DE equivalent; (3) the ablation study confirms that the centroid refinement operator contributes the largest individual improvement component and exhibits a superadditive interaction with adaptive parameter control; and (4) AHCF achieves these quality improvements with an empirical computational overhead of 11.8× relative to standard K-Means — substantially less than the theoretical upper bound and manageable for datasets up to moderate size on standard hardware.

The centroid refinement operator is identified as the single most impactful contribution, providing a principled mechanism for maintaining geometrically valid centroid positions throughout the evolutionary search — a property that is particularly critical in high-dimensional spaces where the curse of dimensionality renders unconstrained DE mutations largely uninformative. Future work will focus on validation on diverse real-world high-dimensional datasets, automatic cluster number determination, and formal theoretical convergence analysis.

BIBLIOGRAPHY

Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In J. Van den Bussche & V. Vianu (Eds.), Database theory — ICDT 2001, Lecture Notes in Computer Science (Vol. 1973, pp. 420–434). Springer. https://doi.org/10.1007/3-540-44503-X_27

Ahmad, I., & Hashim, F. A. (2022). An adaptive differential evolution—particle swarm optimisation approach for clustering problems. Engineering Applications of Artificial Intelligence, 109, 104629. https://doi.org/10.1016/j.engappai.2021.104629

Aloise, D., Deshpande, A., Hansen, P., & Popat, P. (2009). NP-hardness of Euclidean sum-of-squares clustering. Machine Learning, 75(2), 245–248. https://doi.org/10.1007/s10994-009-5103-0

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. Pattern Recognition, 46(1), 243–256. https://doi.org/10.1016/j.patcog.2012.07.021

Arthur, D., & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (pp. 1027–1035). SIAM.

Bellman, R. E. (1961). Adaptive control processes: A guided tour. Princeton University Press.

Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When is 'nearest neighbor' meaningful? Proceedings of ICDT 1999, Lecture Notes in Computer Science (Vol. 1540, pp. 217–235). Springer. https://doi.org/10.1007/3-540-49257-7_15

Calinski, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. Communications in Statistics, 3(1), 1–27. https://doi.org/10.1080/03610927408827101

Celebi, M. E., Kingravi, H. A., & Vela, P. A. (2013). A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Systems with Applications, 40(1), 200–210. https://doi.org/10.1016/j.eswa.2012.07.021

Das, S., Abraham, A., Chakraborty, U. K., & Konar, A. (2008). Differential evolution using a neighborhood-based mutation operator. IEEE Transactions on Evolutionary Computation, 13(3), 526–553. https://doi.org/10.1109/TEVC.2008.2009457

Das, S., & Suganthan, P. N. (2011). Differential evolution: A survey of the state-of-the-art. IEEE Transactions on Evolutionary Computation, 15(1), 4–31. https://doi.org/10.1109/TEVC.2010.2059031

Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1(2), 224–227. https://doi.org/10.1109/TPAMI.1979.4766909

Ezugwu, A. E., Ikotun, A. M., Oyelade, O. O., Abualigah, L., Agushaka, J. O., Eke, C. I., & Akinyelu, A. A. (2022). A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. Engineering Applications of Artificial Intelligence, 110, 104743. https://doi.org/10.1016/j.engappai.2022.104743

Hancer, E., Xue, B., & Zhang, M. (2020). A survey on feature selection approaches for clustering. Artificial Intelligence Review, 53(6), 4519–4545. https://doi.org/10.1007/s10462-019-09800-w

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. Pattern Recognition Letters, 31(8), 651–666. https://doi.org/10.1016/j.patrec.2009.09.011

Lloyd, S. (1982). Least squares quantization in PCM. IEEE Transactions on Information Theory, 28(2), 129–137. https://doi.org/10.1109/TIT.1982.1056489

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

Nanda, S. J., & Panda, G. (2014). A survey on nature inspired metaheuristic algorithms for partitional clustering. Swarm and Evolutionary Computation, 16, 1–18. https://doi.org/10.1016/j.swevo.2013.11.003

Özer, A. B., & Aydin, I. (2023). Hybrid metaheuristic algorithms for K-means-based color image segmentation. Expert Systems with Applications, 214, 119164. https://doi.org/10.1016/j.eswa.2022.119164

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

Peng, C., Wu, X., Yuan, H., & Wang, C. (2021). A modified differential evolution algorithm for high-dimensional clustering optimization. Applied Soft Computing, 100, 106962. https://doi.org/10.1016/j.asoc.2020.106962

Price, K., Storn, R. M., & Lampinen, J. A. (2005). Differential evolution: A practical approach to global optimization. Springer.

Qin, A. K., Huang, V. L., & Suganthan, P. N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Transactions on Evolutionary Computation, 13(2), 398–417. https://doi.org/10.1109/TEVC.2008.927706

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20, 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

Storn, R., & Price, K. (1997). Differential evolution — a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, 11(4), 341–359. https://doi.org/10.1023/A:1008202821328

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9, 2579–2605.

Wang, Y., Chen, X., Fang, H., & Gu, B. (2024). Deep embedding clustering with adaptive autoencoders for high-dimensional data. IEEE Transactions on Neural Networks and Learning Systems, 35(1), 871–884. https://doi.org/10.1109/TNNLS.2022.3183086

Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. Annals of Data Science, 2(2), 165–193. https://doi.org/10.1007/s40745-015-0040-1

Zaharie, D. (2009). Influence of crossover on the behavior of Differential Evolution Algorithms. Applied Soft Computing, 9(3), 1126–1138. https://doi.org/10.1016/j.asoc.2009.02.012

Zhang, J., & Sanderson, A. C. (2009). JADE: Adaptive differential evolution with optional external archive. IEEE Transactions on Evolutionary Computation, 13(5), 945–958. https://doi.org/10.1109/TEVC.2009.2014613