Received: 03 July 2023 | *Revised:* 27 August 2023 | *Accepted:* 02 September 2023
DOI: 10.37917/ijeee.20.1.25 *Vol. 20 | Issue 1 | June 2024*

Iraqi Journal for Electrical and Electronic Engineering
Original Article

Open Access
IJEEE University of Basrah College of Engineering

A Comparative Evaluation of Initialization Strategies for K-Means Clustering with Swarm Intelligence Algorithms

Athraa Qays Obaid, Maytham Alabbas*
Department of Computer Science, College of Computer Science and Information Technology, University of Basrah, Basrah, Iraq

Correspondance
*Maytham Alabbas
Department of Computer Science,
College of Computer Science and Information Technology,
University of Basrah, Basrah, Iraq
Email: ma@uobasrah.edu.iq

Abstract
Clustering is a fundamental data analysis task that presents challenges. Choosing proper initialization centroid techniques is critical to the success of clustering algorithms, such as k-means. The current work investigates six established methods (random, Forgy, k-means++, PCA, hierarchical clustering, and naive sharding) and three innovative swarm intelligence-based approaches—Spider Monkey Optimization (SMO), Whale Optimization Algorithm (WOA) and Grey Wolf Optimizer (GWO)—for k-means clustering (SMOKM, WOAKM, and GWOKM). The results on ten well-known datasets strongly favor swarm intelligence-based techniques, with SMOKM consistently outperforming WOAKM and GWOKM. This finding provides critical insights into selecting and evaluating centroid techniques in k-means clustering. The current work is valuable because it provides guidance for those seeking optimal solutions for clustering diverse datasets. Swarm intelligence, especially SMOKM, effectively generates distinct and well-separated clusters, which is valuable in resource-constrained settings. The research also sheds light on the performance of traditional methods such as hierarchical clustering, PCA, and k-means++, which, while promising for specific datasets, consistently underperform swarm intelligence-based alternatives. In conclusion, the current work contributes essential insights into selecting and evaluating initialization centroid techniques for k-means clustering. It highlights the superiority of swarm intelligence, particularly SMOKM, and provides actionable guidance for addressing various clustering challenges.

Keywords
Clustering, K-means, Centroid Initialization, Swarm Intelligence, Performance Evaluation.

I. Introduction

Clustering is a popular technique used in many data mining and machine learning applications for grouping similar data points into clusters [1]. One of the most widely used clustering algorithms is K-means, which partitions the data into k clusters by iteratively assigning each data point to the nearest centroid and updating the centroids based on the mean of the assigned points [2]. However, despite its effectiveness in many scenarios, K-means clustering has several significant limitations [3]: (i) It is sensitive to the initial placement of centroids, making it prone to suboptimal solutions; (ii) It assumes that clusters have spherical shapes, limiting its ability to handle complex data structures; (iii) Predefining the number of clusters can be difficult in real-world applications; (iv) K-means is sensitive to outliers, which can distort clustering results; and (v) It is not well-suited for categorical or non-numeric data. Many initialization techniques have been proposed to address the first limitation, including random initialization, K-means++ initialization, and hierarchical clustering. In recent years, swarm intelligence algorithms have emerged as a promising alternative for optimizing the initialization of cluster centroids in K-means clustering.

The main objective of this paper is to explore the use of swarm intelligence algorithms for optimizing K-means initialization and investigate four research questions:

- RQ1: How can swarm intelligence algorithms be used to optimize the initialization of cluster centroids in K-means clustering?
- RQ2: Can swarm algorithms effectively initialize cluster centroids for K-means compared to traditional algorithms?
- RQ3: Which swarm intelligence algorithms are most effective for selecting the optimal initial centroids in K-means clustering?
- RQ4: How does the choice of swarm intelligence algorithm affect the performance of K-means clustering concerning centroid initialization?

The paper is structured as follows: Section II. covers related works; Section III. discusses the theoretical background, including the K-means algorithm with common initialization centroid techniques and swarm intelligence algorithms; Section IV. presents the current work; Section V. displays the results and provides a discussion; and Section VI. concludes the paper.

II. Related Works

Numerous initialization methods have been proposed to address the challenges of the K-means algorithm. However, their results have been somewhat limited. This section presents an overview of the research conducted in this domain.

In [4], four initialization methods for the K-means algorithm were compared: random, Forgy, MacQueen, and Kaufman. The authors conducted experiments to draw up the probability distribution of the square-error values of the final clusters returned by the K-means algorithm, independently of the initial clustering and the order of the instances when each of the four initialization methods was used. The results showed that the random and Kaufman initialization methods outperformed the other methods, making K-means more effective and less dependent on the initial clustering and the order of the instances. The Kaufman initialization method also induced a more desirable behavior with respect to the convergence speed than the random initialization method.

The work in [5] presented a deterministic initialization method for K-means clustering called PCA-Part. The method was based on PCA and the divisive hierarchical approach. PCA-Part led K-means to generate clusters with the sum of squared error (SSE) values close to the minimum SSE values obtained by 100 random start runs. Furthermore, PCA-Part often led K-means to faster convergence compared to random methods.

In [6], the work introduced modifications to Var-Part and PCA-Part, deterministic and linear hierarchical K-means initialization methods. The modified versions were compared to popular linear methods such as Forgy's, MacQueen's, Maximin, and K-means++, using diverse UCI Machine Learning Repository data sets. Despite being deterministic, results revealed that Var-Part and PCA-Part were highly competitive with K-means++, one of the best random initialization methods. The proposed modifications significantly enhanced the performance of both hierarchical methods. These modified variants of Var-Part and PCA-Part offered effective initialization for K-means, particularly in time-sensitive applications with large data sets, and could also function as approximate clustering algorithms without the need for subsequent K-means refinement.

The authors in [7] provided an overview of various initialization methods for the K-means clustering algorithm, focusing on their computational efficiency. They compared eight commonly used linear time complexity methods using diverse datasets and performance criteria. Through non-parametric statistical tests, the analysis demonstrated that popular methods such as Forgy, MacQueen, and Maximin frequently produced poor results. The paper highlighted the existence of significantly better alternatives with comparable computational requirements. Overall, the findings offered recommendations for practitioners, suggesting more effective initialization methods for K-means clustering.

In [8], the work addressed the sensitivity of K-means clustering to initial centroid selection and proposed a selection method for improved performance. The proposed method began by randomly selecting initial centroids and evaluating their suitability based on distance calculations, i.e., Euclidean and Manhattan, with other data points in the dataset. The study evaluated the proposed initialization method on various datasets with different characteristics and complexities. Experimental results demonstrated that the proposed method was more effective and yielded more accurate clustering than random initialization. The work concluded by highlighting the significance of good clustering algorithms and emphasized the practicality and efficiency of the proposed initialization method, which outperformed standard K-means in terms of both speed and accuracy.

The work in [9] discussed the performance of three classical K-means initialization strategies: Random Partition Method, K-means++, and PCA-based K-means. The experiment evaluated their performance on the UCI Machine Learning handwritten digits dataset regarding runtime and clustering quality. The results showed that all three strategies found similar cluster centroids with comparable clustering accuracy. However, the PCA-based K-means strategy significantly improved the running time and outperformed the other strategies.

In [10], the authors presented improvements in the performance of the rough k-means clustering algorithm by proposing a new initialization algorithm, a new performance measure,

and a new method for selecting the zeta value. The results show that the proposed algorithm outperforms the existing ones on various datasets regarding S/O index, RMSSTD, and computational complexity. However, the proposed work does not consider the impact of the weights of lower and upper approximations, and it only applies to the Peters refined rough k-means algorithm.

In [11], the authors conducted a critical and experimental analysis of different variants of the k-means algorithm on six benchmark datasets. The results indicate that no single solution exists for the problems of the k-means algorithm and that each variant is either data-specific or application-specific.

In [12], the work discussed the different initialization methods for the K-means clustering algorithm and their effects on its performance. The K-means algorithm is widely used but has limitations such as sensitivity to outliers and reliance on data features. Various initialization techniques have been proposed to overcome these issues. The study compared methods like Random, K-means++, Maximin, Robust Initialization (ROBIN), Kaufman, and DK-means++ to investigate their impact on K-means variations. It showed that sophisticated initialization methods can reduce performance differences among K-means implementations, and deterministic methods like DK-means++ can achieve better average performance. However, stochastic methods may perform better if executed multiple times.

In [13], the authors proposed an entropy-based initialization method for the K-means clustering algorithm and a method to determine the optimal number of clusters. The proposed methods achieve better clustering results with faster convergence and lower computational cost than other methods. However, there are some limitations, such as dependence on the threshold value for the entropy-based initialization and difficulty choosing the best cluster validity index for different data sets.

In [14], the authors proposed a method for selecting initial cluster centers for K-means. The method finds outer points using a Convex Hull, selects the farthest points as initial centroids, and discards nearest neighbors to avoid choosing from the same cluster. The method outperforms conventional K-means and other methods regarding clustering error, computation time, and Cluster Compactness and Separation Index (CCPI) for four real-world datasets. However, it is sensitive to outliers and requires a parameter.

In [15], the authors proposed a new method, BRIk, to initialize the k-means clustering algorithm. BRIk uses bootstrapped replications of the data and randomly initialized k-means to obtain a set of centroids. These centroids are then clustered again, and the deepest point in each cluster is chosen as an initial seed. BRIk performs well in minimizing distortion and recovering the true cluster structure, especially for complex datasets with high dimensions and many clusters. However, BRIk has a higher computational cost than some of the other methods.

In [16], the authors compared 17 k-means initialization algorithms on 6,000 synthetic and 28 real-world datasets. The results show that no single algorithm outperforms all others in all cases, and the performance depends on various factors, such as the data distribution and the number of clusters. The limitations of the proposed work are that it does not consider other aspects of clustering quality, such as the interpretability of the clusters, and it does not explore the impact of data pre-processing or parameter tuning on the algorithms. Below is a summary of the related works in Table I.

III. Materials and Methods

A. The K-means clustering

One of the most commonly used clustering techniques is K-means clustering. It has numerous applications in computer vision, pattern recognition, and information retrieval. Its main objective is to group similar data points by partitioning the input dataset of n points into k clusters [17]. The process involves assigning each point to the cluster with the nearest centroid through an iterative process. The centroid of each cluster is then recalculated by calculating its mean. Algorithm 1 illustrates the pseudocode of K-means clustering [3].

Algorithm 1 K-means clustering

1: An initial clustering is created by choosing k random centroids from the dataset.
2: For each data point, calculate the distance from all centroids and assign its membership to the nearest centroid.
3: Recalculate the new cluster centroids by the average of all data points that are assigned to the clusters.
4: Repeat steps 2-3 until convergence.

Selecting initial centroids for K-means clustering is a challenging task, and it can be approached in various ways using common initialization methods. These methods include random initialization, Forgy initialization, K-means++ initialization, initialization based on hierarchical clustering, and initialization based on prior knowledge or domain expertise. Each method has its own factors to consider, and the choice of initialization method depends on the specific dataset and problem at hand. Experimenting with multiple initialization methods is often recommended to determine the most suitable one for achieving optimal clustering results. The initialization step plays a crucial role in the K-means algorithm, setting the starting point for the iterative optimization process.

Here are some concise explanations of the most frequently used methods for choosing initial centroids in K-means clus-

tering:

1) Random Technique

This technique randomly selects K data points from the dataset as the initial centroids. It is easy to implement but may lead to suboptimal clusters if the initial centroids do not accurately represent the true cluster centers [4].

2) Forgy Technique

This technique is similar to random initialization, but instead of randomly selected data points, it samples K data points directly from the dataset. This technique can improve upon random initialization by ensuring that the initial centroids are representative of the data distribution. However, it can still suffer from limitations associated with selecting arbitrary initial centroids, such as the possibility of converging to a suboptimal clustering solution [2].

3) Initialization Based on Hierarchical Clustering

This technique involves performing hierarchical clustering on the dataset and then selecting K clusters at an appropriate level of the dendrogram. The centroids of these clusters are then used as the initial centroids for the K-means algorithm. This technique can provide a promising starting point for optimization by leveraging the hierarchical structure of the data [8].

4) K-means++ Technique

This technique has been developed to overcome the limitations of random and Forgy initialization methods. It involves selecting initial centroids evenly distributed throughout the dataset, achieved by giving higher probabilities to data points further away from existing centroids. This ensures that the chosen centroids reflect the overall distribution of the data [12].

5) Initialization Based on PCA

This technique involves applying PCA (Principal Component Analysis) to the dataset. PCA identifies the most informative features in the dataset by transforming the data into a lower-dimensional space that preserves the most variance. The clustering algorithm is then initialized by selecting K data points from the reduced-dimensional space. This method aims to capture the significant sources of variation in the data by considering the principal components, potentially leading to improved clustering results [18–20].

6) Naive sharding Technique

It is a technique for partitioning a dataset into multiple shards or subsets without considering the underlying data distribution. It typically divides the data evenly into fixed-size shards, regardless of the data characteristics. This simplistic approach may not consider the data's clustering structure and can result in imbalanced shards, leading to suboptimal clustering performance [21].

B. Swarm Intelligence Algorithms

This section will cover three commonly used swarm intelligence algorithms: Grey Wolf Optimizer (GWO), Spider Monkey Optimization (SMO), and Whale Optimization Algorithm (WOA). These algorithms were chosen over other optimization algorithms due to their natural inspiration and demonstrated efficacy in addressing diverse optimization challenges, particularly those involving numerous variables or significant nonlinearity and relatively newer status. While particle swarm optimization (PSO), artificial bee colony (ABC), and genetic algorithm (GA) are also established and effective, GWO, SMO, and WOA have exhibited superior global solution-finding efficiency. The optimal algorithm selection depends on problem characteristics, but all three – GWO, SMO, and WOA – are strong contenders for various optimization problems.

1) GWO

GWO is a population-based metaheuristic algorithm inspired by the leadership hierarchy and hunting behavior of grey wolves. GWO consists of four types of wolves: alpha, beta, delta, and omega, which represent the four best solutions within the population. The algorithm begins by initializing a population of solutions randomly. During each iteration, the wolves adjust their positions by referencing the positions of the alpha, beta, delta, and omega wolves. The termination of the algorithm is determined by meeting a predetermined stopping criterion. Algorithm 2 illustrates the pseudocode of the GWO algorithm [22].

Algorithm 2 GWO algorithm

```
Initialize the grey wolf population X_i (i = 1, 2, ..., n)
Initialize a, A, and C
Calculate the fitness of each search agent
X_α = the best search agent
X_β = the second-best search agent
X_δ = the third-best search agent
while (t < Max number of iterations) do
    for each search agent do
        Update the position of the current search agent
    end for
    Update a, A, and C
    Calculate the fitness of all search agents
    Update X_α, X_β, and X_δ
    t = t + 1
end while
return X_α
```

TABLE I.
RELATED WORKS

Ref	Year	Technique(s)	Dataset(s)	Results
[4]	1999	• Random • Forgy approach • Macqueen approach • Kaufman approach	• Iris • Ruspini • Glass	• The random and Kaufman initialization methods are more effective and robust than the Forgy and MacQueen initialization methods when used with the K-means algorithm. • The Kaufman initialization method exhibits a more desirable behavior with respect to convergence speed than the random initialization method when used with the K-means algorithm.
[5]	2004	• PCA-Part • Random seed • Random partition	• Pendigits • Segmentation • Letter	• PCA-Part is a promising initialization method for K-means clustering that often leads to significantly faster convergence and significantly lower SSE values. • Further research is needed to explore other ways of partitioning the sample space or combining random and deterministic restarts for initializing K-means.
[6]	2012	• Forgy method • MacQueen • Maximin method • K-means++ method • PCA-Part method • Var-Part method • Modification to Var-part and PCA-part	• Abalone • Breast Cancer Wisconsin (Original) • Breast Tissue • Ecoli • Glass Identification • Heart Disease • Ionosphere • Iris (Bezdek) • ISOLET • Landsat Satellite (Statlog) • Letter Recognition • MAGIC Gamma Telescope • Multiple Features (Fourier) • Musk (Clean2) • Optical Digits • Page Blocks Classification • Pima Indians Diabetes • Shuttle (Statlog) • Spambase • SPECTF Heart • Wall-Following Robot Navigation • Wine Quality • Wine • Yeast	• The paper proposes a modification to two hierarchical K-means initialization methods, Var-Part and PCA-Part, using Otsu's method, which significantly improves their performance. • The modified methods can be used effectively in time-critical applications with large data sets.
[7]	2013	• **Linear time complexity** o Forgy method o Jancey method o MacQueen o Ball and Hall method o Simple Cluster Seeking method	• Breast cancer wisconsin (original) • Cloud cover (DB1) • Concrete compressive strength • Corel image features • Covertype • Ecoli	• Eight linear-time initialization methods for the K-means clustering algorithm were compared on a large and diverse collection of real and synthetic data sets. • The study demonstrated that popular initialization methods often perform poorly and that there are strong alternatives to these methods.

		o Spath method o Maximin method o Al-Daoud density-based method o Bradley and Fayyad method o K-means++ method o PCA-Part method o The Var-Part method o Lu et al.'s method o Onoda et al.'s method • **Loglinear time-complexity** o Hartigan's method o Al-Daoud variance-based method o Redmond and Heneghan method o The ROBIN (ROBust INitialization) method • **Quadratic-complexity** o Astrahan method o Lance and Williams o Kaufman and Rousseeuw method o Cao, Liang, and Jiang • **Other methods** o Binary-splitting method o Directed-search binary-splitting method o Global K-means method	• Steel plates faults • Glass identification • Heart disease • Ionosphere • ISOLET • Landsat satellite (Statlog) • Letter recognition • MAGIC gamma telescope • Multiple features (Fourier) • MiniBooNE particle identification • Musk (Clean2) • Optical digits • Page blocks identification • Parkinsons • Pen digits • Person activity • Pima Indians diabetes • Image segmentation • Shuttle (Statlog) • SPECTF heart • Telugu vowels • Vehicle silhouettes (Statlog) • Wall-following robot navigation • Wine quality • World TSP • Yeast	
[8]	2013	• Random method • DIMK-means (Distance-based Initialization Method for K-means)	• **Artificial datasets** o Ruspini o Rfivec • **Real datasets** o IRIS o Wine recognition • **Libras Movement**	• The proposed method improves the K-means algorithm by reducing its sensitivity to initial centroid selection, resulting in more accurate and consistent clustering results. • The proposed method outperforms random selection in terms of speed, accuracy, stability, and reliability on different datasets and measures.
[9]	2018	• Random Partition method • K-means++ method • PCA-based K-means	• Hand-written digits	• The study compares three K-means initialization strategies on the UCI machine learning handwritten digits dataset. • The study finds that the PCA-based K-means strategy is significantly faster than the other two strategies and produces clustering results with similar accuracy.

[10]	2020	• New centroids initialization algorithm for rough k-means	• Synthetic • Forest cover • Microarray	• The proposed algorithm outperforms the existing ones on various datasets in terms of S/O index, RMSSTD, and computational complexity. • The proposed work does not consider the impact of the weights of lower and upper approximations, and it only applies to the Peters refined rough k-means algorithm.
[11]	2020	• Variants of the k-means algorithms	• Cleveland Heart Disease • KDD-Cup 1999 (10%) • Wisconsin Diagnostic Breast Cancer • Epileptic Seizure Recognition • Credit Approval • Postoperative	• The results indicate that no single solution exists for the problems of the k-means algorithm and that each variant is either data-specific or application-specific.
[12]	2021	• Random method • K-means++ method • Maximin method • Kaufman method • ROBIN method • Density K-means++ (DK-means++)	• Iris • Ionosphere • Wine • Breast cancer • Glass • Yeast	• More sophisticated initialization techniques reduce the difference in performance among the K-means variations. Deterministic methods perform better than stochastic methods on average. • Stochastic methods can achieve better clustering performance if executed multiple times. However, deterministic methods can still be competitive for large data sets where execution time is a factor.
[13]	2021	• Maximization of Shannon's entropy of the data distribution (initial points) • Four cluster validity indexes: partition coefficient, classification entropy, separation index, and partition index (select K)	• synthetic data • real-life data	• The proposed methods can achieve better clustering results with faster convergence and lower computational cost than other methods. • The proposed methods have some limitations, such as dependence on the threshold value for the entropy-based initialization and difficulty choosing the best cluster validity index for different data sets.
[14]	2021	• Convex Hull algorithm	• Synthetic • Iris • Wine • Letter • Ruspini	• It outperforms the conventional K-means and other existing methods regarding clustering error, computation time, and CCPI for four real-world datasets. Also, it performs well on a synthetic dataset with six clusters. • It is sensitive to outliers, requires a parameter that may vary depending on the data distribution and may not work well when the number of clusters is two.
[15]	2021	• BRIk: Bootstrap replications and performs Randomly Initialized k-means	• **Simulated (synthetic data)** • **Real data** o Breast cancer diagnostic (BC) o Breast tissue (BT) o Ecoli (EC) o Forest types (FT) o Glass identification (GI)	• BRIk effectively minimizes distortion and recovers true cluster structure, especially for complex datasets. • BRIk is computationally expensive, but this can be reduced by using a small bootstrap size or running it multiple times.

			o Heart disease-Hung. Reprocessed (HD) o Hill valley-training (HV) o Image segmentation (IS) o Ionosphere (I) o Libras movement (LM) o Multiple features (MF) o Page blocks classification (PBC) o Parkinson (P) o Pima Indians diabetes (PID) o Spambase (SB) o Steel plates faults (SPF) o Synthetic control chart (SCC) o Vertebral column (VC) o Wine (W) o Wine quality red (WQ)	
[16]	2022	• Random • Continuous K Means (Ck-Means) • Milligan (Milligan) • Katsavounidis, Kuo & Zhang (Kkz) • Bradley & Fayyad (Bf) • Global K-Means (Gkm) • Yuan Et Al. (Yuan) • Hand & Krzanowski (Hk) • Intelligent K-Means (Ik_1 And Ik_2) • K-Means++ (Km++) • Single Pass Seed Selection (Spss) • Erisoglu, Calis & Sakallioglu (Ecs) • Hatamlou Binary Search (Bs) • Khan's Seed Selection Algorithm (Khan) • Onoda, Sakai & Yamada (Osy_1 And Osy_2)	• Synthetic • real-world o Avila o Blood Transfusion o Breast Cancer (Diag.) o Breast Cancer (Orig.) o Breast Tissue o Ecoli o Fossil o Glass o HTRU2 o Haberman o Iris o Leaf o Letter Recognition o Libras Movement o Musk 1 o Musk 2 o Optical Recognition o Page Blocks o Parkinsons o Pen-Based Recognition o Sonar all o Spambase o Vehicle Silhouettes o Vertebral Column o Wine o Wine Quality (Red) o Wine Quality (White) o Yeast	• No single k-means initialization algorithm outperforms all others in all cases. The performance depends on various factors, such as the data distribution and the number of clusters. • The proposed work does not consider other aspects of clustering quality, such as the interpretability of the clusters, and it does not explore the impact of data pre-processing or parameter tuning on the algorithms.

2) SMO

SMO is a nature-inspired metaheuristic algorithm that mimics the foraging and communication strategies of spider monkeys to solve optimization problems. SMO utilizes a population of solutions and uses various operators such as exploration, exploitation, and information sharing to navigate the search space effectively. It demonstrates competitive performance compared to other metaheuristic algorithms across benchmark functions [23]. SMO offers a promising approach for addressing complex optimization tasks and has potential applications in diverse real-world domains. Algorithm 3 illustrates the pseudocode of the SMO algorithm [24].

Algorithm 3 SMO algorithm

Initialize Population, LocalLeaderLimit, GlobalLeaderLimit, pr.
Calculate fitness
Select global leader and local leaders by applying greedy selection.
while (Termination criteria is not satisfied) **do**
    For finding the objective (Food Source), generate the new positions for all the group members by using self experience, local leader experience and group members experience.
    Apply the greedy selection process between existing position and newly generated position, based on fitness and select the better one.
    Calculate the probability probi for all the group members.
    Produce new positions for the all the group members, selected by probi, by using self experience, global leader experience and group members experiences.
    Update the position of local and global leaders, by applying the greedy selection process on all the groups.
    If any Local group leader is not updating her position after a specified number of times (LocalLeaderLimit) then re-direct all members of that particular group for foraging.
    If Global Leader is not updating her position for a specified number of times (GlobalLeaderLimit) then she divides the group into smaller groups.
end while

3) WOA

WOA is a metaheuristic algorithm inspired by the hunting behavior of whales. It utilizes three key operators: encircling prey, bubble-net feeding, and searching for prey. The algorithm starts by initializing a population of whales randomly. In each iteration, the whales update their positions based on the equations derived from the operators. The algorithm terminates when a specified stopping criterion is met, indicating the discovery of an optimal solution. Algorithm 4 illustrates the pseudocode of the WOA algorithm [25].

Algorithm 4 WOA algorithm

Initialize the whale's population X_i (i = 1, 2, ..., n)
Calculate the fitness of each search agent
X^* = the best search agent
while (t < maximum number of iterations) **do**
    for each search agent **do**
        Update a, A, C, l, and p
        if ($p < 0.5$) **then**
            if ($|A| < 1$) **then**
                Update the position of the current search agent
            else if ($|A| \geq 1$) **then**
                Select a random search agent (X_{rand})
                Update the position of the current search agent
            end if
        else if ($p \geq 0.5$) **then**
            Update the position of the current search
        end if
    end for
    Check if any search agent goes beyond the search space and amend it
    Calculate the fitness of each search agent
    Update X^* if there is a better solution
    t = t + 1
end while
return X^*

C. Clustering Evaluation

To assess the clustering outcome, the Silhouette coefficient is utilized. The silhouette coefficient measures the quality and separation of clusters in a clustering analysis. It is calculated for each data point and represents the cohesion within its cluster and the separation from neighboring clusters. The silhouette coefficient is calculated using Eq. 1.

$$Silhouette_i = \frac{(b_i - a_i)}{max(a_i, b_i)}, \tag{1}$$

where $Silhouette_i$ is the silhouette coefficient for data point i, a_i is the average dissimilarity between i and other data points in the same cluster, and b_i is the minimum average dissimilarity between i and any other cluster. $Silhouette_i$ ranges from -1 to 1. A value close to 1 means well-clustered data, 0 suggests data on the boundary, and -1 implies possible misassignment of data to clusters [9].

IV. The Current Work

In the current work, we employed three swarm intelligence algorithms, namely GWO, SMO, and WAO, to address the challenge of initializing centroids in the K-means clustering algorithm. The subsequent lines present a detailed overview of the specific aspect of the current study that is aimed to address the first research question (RQ1).

A. Individual Representation

There are two techniques for representing initial centroids as individuals in swarm algorithms. The indirect technique represents an individual's position by the indices of data points in the dataset. Each index corresponds to a specific sample. The direct technique represents an individual's position by the attributes of data points in the dataset. Each position corresponds to a distinct sample. In this study, we focus primarily on the direct technique due to its superior performance compared to the indirect technique [26].

In this study, each individual in the population is represented by a 2D matrix with K rows (number of clusters) and D columns (number of attributes), representing the data points of the dataset as initial centers for the problem. Here is an example of an individual with 3 clusters and 4 attributes:

Centers	Attribute 1	Attribute 2	Attribute 3	Attribute 4
Cluster 1	6.7	3.1	5.6	2.4
Cluster 2	6.2	3.4	5.4	2.3
Cluster 3	5.0	3.4	1.5	0.2

In this representation, each row corresponds to a cluster center, and the columns contain the attribute values of the respective centroids.

B. Objective Function

To achieve a balance between clustering quality and computational speed, a two-part objective function is proposed. The first part quantifies the clustering quality by transforming the silhouette coefficient into a minimization problem by subtracting it from 1. The second part evaluates the computational speed by employing the number of iterations required to obtain the results. The objective function is computed using Eq. 2, which combines these two aspects in a unified framework. By optimizing this objective function, a trade-off between clustering performance and computational efficiency can be achieved.

$$f(ind) = a \times (1 - Silhouette(KM(ind))) + b \times \frac{iterations}{Max\_iter}, \quad (2)$$

where f is the objective function, ind is an individual, KM is the K-means algorithm, a and b are real numbers within the interval [0, 1], and $a+b=1$. They are coefficients that represent the weights assigned to the clustering quality and computational speed, respectively. The value of a determines how much weight is given to the clustering quality in determining the value of f, while the value of b determines how much weight is given to the computational speed. These coefficients can be adjusted to change the relative importance of these two measures in the equation and to achieve the desired values of f.

A lower objective function value, f, signifies improved clustering quality with enhanced cluster distinctiveness and cohesion. Additionally, it indicates faster convergence.

C. Systems

We have investigated three different swarm intelligence-based systems:

1) GWOKM

In this system, the initial centroids for the KM algorithm are selected using a GWO algorithm, as shown in Algorithm 2. The system employs the individual representation described earlier and evaluates the clustering quality using the objective function outlined in Eq. 2.

2) SMOKM

This system is similar to GWOKM, except that it employs the SMO algorithm, as shown in Algorithm 3, as the optimization algorithm instead of the GWO algorithm.

3) WOAKM

This system is similar to GWOKM, except that it employs the WOA algorithm, as shown in Algorithm 4, as the optimization algorithm instead of the GWO algorithm.

V. Results

A. Parameters Setting

For all experiments conducted in this study, the parameter settings specified in Table II were used consistently for all swarm intelligence algorithms.

TABLE II.
Parameters Setting

Parameters	Value
Population Size	20
Maximum Iterations	100
Number of Runs	10

B. Tested Datasets

To assess the effectiveness of the current study, ten well-known real datasets were used to evaluate the proposed approaches, compared with other techniques. The characteristics of these datasets are provided in Table III.

TABLE III.
DESCRIPTIONS OF TESTED DATASETS

ID	Dataset	Numbers× Attributes	Number of clusters
1	Glass	214 × 9	6
2	Bupa	345 × 6	2
3	Seed	210 × 7	3
4	Iris	150 × 4	3
5	Breast-Cancer	569 × 30	2
6	Mall-Customers	200 × 5	5
7	Digits	1797 × 64	10
8	Heart	270 × 13	2
9	Haberman	306 × 3	2
10	CMC	1473 × 9	3

C. Results

In this work, we investigated the performance of six initialization centroid techniques for K-means clustering: Random, PCA, Forgy, K-means++, Naive sharding, and Hierarchical Clustering (see Section III (A)). We also investigated the performance of three proposed swarm intelligence techniques: SMOKM, WOAKM, and GWOKM (see Section IV.). We evaluated these techniques on ten diverse real-world datasets: Glass, Bupa, Seed, Iris, Breast-Cancer, Mall-Customers, Digits, Heart, Haberman, and CMC (see Table III). Our objective is to evaluate the clustering performance and computational efficiency of these techniques to identify the most effective approaches on different real-world datasets. The results of these experiments, in terms of (mean ± standard deviation), for ten runs, are summarized in Table IV.

D. Discussion

The results presented in TableIV provide valuable insights into the effectiveness and performance of various centroid initialization techniques for K-means clustering. These results offer insights into how different initialization strategies impact the quality of clustering results. By examining the measures of clustering quality and computational speed presented in the table, we can gain a deeper understanding of how these techniques perform and their implications for clustering algorithms. Let us now discuss the results in detail to uncover patterns, trends, and noteworthy observations.

For the Glass dataset, the SMOKM technique stood out as the superior option, achieving the highest average silhouette coefficient of 0.592 ± 0.001. This indicates that SMOKM produced well-defined and separated clusters, resulting in better clustering performance. Moreover, SMOKM demonstrated a relatively fast convergence rate, requiring an average of 3.0 iterations. Considering both the high silhouette coefficient and the efficient convergence rate, SMOKM emerged as the preferable choice for clustering the Glass dataset.

Similarly, for the Bupa dataset, SMOKM, WOAKM, and GWOKM techniques exhibited the highest average silhouette coefficients with a minimal standard deviation of 0.634 ± 0.0, indicating consistent and well-separated clusters. Among these techniques, SMOKM demonstrated the fastest convergence rate with an average of 2.0 iterations. While all three techniques achieved comparable clustering performance, SMOKM's combination of high quality and computational efficiency made it the preferred option for clustering the Bupa dataset.

Moving on to the Seed dataset, SMOKM and WOAKM techniques showed the highest average silhouette coefficients with low standard deviations of 0.473 ± 0.0, indicating superior clustering performance. SMOKM also exhibited faster convergence, requiring an average of 2.3 iterations compared to WOAKM's 3.3 iterations. Thus, SMOKM proved effective in producing well-separated clusters and offered a faster computational speed, making it the preferred choice for clustering the Seed dataset.

In the case of the Iris dataset, several techniques, including PCA, Hierarchical Clustering, SMOKM, WOAKM, and GWOKM, yielded the highest average silhouette coefficients of 0.553 ± 0.0 with minimal standard deviation. These techniques consistently produced well-separated clusters, indicating superior clustering performance. Regarding computational efficiency, Hierarchical Clustering and SMOKM stood out with low iteration counts of 2.0 ± 0.0 and minimal standard deviations. Therefore, these techniques provided a favorable balance of clustering performance and computational speed for the Iris dataset.

For the Breast-Cancer dataset, multiple techniques, like Random, PCA, Forgy, K-means++, Naïve sharding, SMOKM, WOAKM, and GWOKM, demonstrated equal silhouette coefficient values of 0.697 ± 0.0, indicating consistent and well-separated clusters. However, when considering the number of iterations required for convergence, SMOKM, WOAKM, and GWOKM exhibited faster convergence rates of 2.0 ± 0.0 compared to other techniques. Therefore, if computational efficiency is a priority, SMOKM, WOAKM, and GWOKM could be preferable due to their faster convergence rates.

TABLE IV.
EXPERIMENTAL RESULTS

Datasets	Techniques																		Best Technique(s)	
	Random		PCA		Forgy		K-means++		Naive Sharding		Hierarchical Clustering		SMOKM		WOAKM		GWOKM		Quality of Clustering	Computational Speed
	Iterations	Silhouette	Iterations	Silhouette	Iterations	Silhouette	Iterations	Silhouette	Iterations	Silhouette	Iterations	Silhouette	Iterations	Silhouette	Iterations	Silhouette	Iterations	Silhouette		
Glass	8.4± 2.289	0.332 ±0.091	4.0± 0.0	0.563 ±0.0	8.4± 2.289	0.332± 0.091	6.9± 4.085	0.452 ±0.045	9.0± 0.0	0.291 ±5.551	3.0± 0.0	0.586 ±0.0	**3.0± 0.0**	**0.592 ±0.001**	3.1± 0.3	0.59± 0.002	3.4± 0.49	0.589 ±0.003	SMOKM	SMOKM
Bupa	9.2± 2.857	0.632 ±0.001	8.0± 0.0	0.631 ±1.11	9.2± 2.857	0.632 ±0.001	5.8± 2.2	0.633 ±0.002	9.0± 0.0	0.631 ±1.11	5.0± 0.0	0.508 ±0.0	**2.0± 0.0**	**0.634 ±0.0**	2.8± 0.4	**0.634 ±0.0**	2.7± 0.458	**0.634 ±0.0**	SMOKM, WOAKM, GWOKM	SMOKM
Seed	9.8± 2.182	0.472± 0.002	15.0 ±0.0	0.469± 5.551	9.8± 2.182	0.472± 0.002	9.8± 2.182	0.472± 0.002	6.0± 0.0	0.473± 5.551	4.0± 0.0	0.473± 5.551	**2.3± 0.458**	**0.473± 0.0**	3.3± 0.9	**0.473 ±0.0**	2.5± 0.5	0.473 ±5.551	SMOKM, WOAKM	SMOKM
Iris	6.4± 2.01	0.54± 0.019	7.0± 0.0	**0.553± 0.0**	6.4± 2.01	0.54± 0.019	7.0± 2.408	0.552± 0.001	5.0± 0.0	0.551 ±0.0	**2.0± 0.0**	**0.553 ±0.0**	**2.0± 0.0**	**0.553 ±0.0**	2.9± 0.3	**0.553 ±0.0**	2.2 ±0.4	**0.553 ±0.0**	PCA, Hierarchical Clustering, SMOKM, WOAKM, GWOKM	Hierarchical Clustering, SMOKM
Breast-Cancer	5.8± 1.327	**0.697± 0.0**	12.0 ±0.0	**0.697± 0.0**	5.8± 1.327	**0.697± 0.0**	5.5 ±3.170	**0.697± 0.0**	5.0± 0.0	**0.697± 0.0**	10.0 ±0.0	0.547 ±0.0	**2.0 ±0.0**	**0.697± 0.0**	**2.0± 0.0**	**0.697± 0.0**	**2.0± 0.0**	**0.697± 0.0**	Random, PCA, Forgy, K-means++, Naïve sharding, SMOKM, WOAKM, GWOKM	SMOKM, WOAKM, GWOKM
Mall-Customers	7.2± 2.315	0.501± 0.058	10.0 ±0.0	0.395± 5.551	7.2± 2.315	0.501± 0.058	5.3± 1.952	0.52± 0.042	13.0 ±0.0	**0.554 ±0.0**	2.0± 0.0	0.468± 5.551	**2.9 ±0.3**	**0.554 ±0.0**	3.1± 0.539	**0.554 ±0.0**	3.2± 0.4	**0.554 ±0.0**	Naive Sharding, SMOKM, WOAKM, GWOKM	SMOKM
Digits	18.9± 6.685	0.176± 0.014	59.5± 13.5	0.178± 0.003	18.9± 6.685	0.176± 0.014	24.1± 7.368	0.18± 0.011	26.0 ±0.0	0.16 ±0.0	8.0 ±0.0	0.126 ±0.0	7.3± 0.458	0.188 ±0.0	8.2± 0.872	0.188 ±0.0	**8.2± 0.98**	**0.189 ±0.001**	GWOKM	GWOKM
Heart	10.0± 3.13	0.378± 0.001	16.0± 0.0	**0.38 ±0.0**	10.0 ±3.13	0.378 ±0.001	9.1± 1.972	0.38± 0.001	9.0 ±0.0	0.377 ±0.0	10.0 ±0.0	0.281 ±5.551	**2.3± 0.458**	**0.38 ±0.0**	2.5± 0.5	**0.38 ±0.0**	2.4± 0.49	**0.38 ±0.0**	PCA, SMOKM, WOAKM, GWOKM	SMOKM
Haberman	8.5± 2.419	0.394 ±0.013	11.0 ±0.0	0.357 ±0.0	8.5± 2.419	0.394 ±0.013	7.5± 2.335	0.393 ±0.012	2.0 ±0.0	0.397 ±0.0	4.0 ±0.0	0.422 ±5.551	**2.0 ±0.0**	**0.401 ±0.0**	2.8 ±0.98	**0.401 ±0.0**	2.5± 0.671	**0.4 ±0.0**	SMOKM, WOAKM, GWOKM	SMOKM
CMC	9.8± 2.96	0.442 ±0.002	14.0 ±0.0	**0.443 ±0.0**	9.8 ±2.96	0.442 ±0.002	8.6± 3.231	0.441 ±0.003	7.0 ±0.0	0.438 ±5.551	4.0 ±0.0	0.439 ±5.551	**3.1 ±0.3**	**0.443 ±0.0**	4.0± 0.632	**0.443 ±0.0**	**3.1 ±0.3**	**0.443 ±0.0**	PCA, SMOKM, WOAKM, GWOKM	SMOKM, GWOKM
No. of times selected as best choice	0	1	0	**4**	0	1	0	1	0	2	1	1	**9**	**9**	1	**8**	3	**8**		

Analyzing the Mall-Customers dataset, Naive sharding, SMOKM, WOAKM, and GWOKM techniques yielded the highest average silhouette coefficients of 0.554 ± 0.0 with minimal standard deviation. These techniques consistently produced well-separated clusters, indicating superior clustering performance. Among them, SMOKM demonstrated the fastest convergence rate of 2.9 ± 0.3, followed by WOAKM and GWOKM. Considering the quality of clustering results and computational speed, SMOKM emerged as the preferred choice for the Mall-Customers dataset.

Regarding the Digits dataset, GWOKM emerged as the better option, achieving the highest average silhouette coefficient of 0.189 ± 0.001. GWOKM produced well-defined and separated clusters, resulting in better clustering performance. Regarding speed, GWOKM required an average of 8.2 iterations, suggesting a relatively efficient convergence process. Therefore, considering the high silhouette coefficient and the relatively efficient convergence rate, GWOKM was the preferable choice for clustering the Digits dataset.

For the Heart dataset, PCA, SMOKM, WOAKM, and GWOKM techniques exhibited the highest average silhouette coefficients of 0.38 ± 0.0 with minimal standard deviation. These techniques consistently produced well-separated clusters, indicating superior clustering performance. Among them, SMOKM demonstrated low iteration counts of 2.3 ± 0.458 with minimal standard deviations, indicating both effective clustering and computational efficiency. Therefore, SMOKM provided a favorable balance of clustering performance and computational speed for the Heart dataset.

Moving on to the Haberman dataset, the results show that Hierarchical Clustering has the highest mean value (0.422) but also the highest standard deviation (5.551). SMOKM, WOAKM, and GWOKM have the same mean value (0.401) and a standard deviation of 0.0. This suggests that Hierarchical Clustering might perform slightly better on average, but its results are more variable and less consistent. SMOKM, WOAKM, and GWOKM, on the other hand, are more consistent and, therefore, better choices due to their lower variability. Moreover, SMOKM exhibited a relatively fast convergence rate, requiring an average of iterations 2.0±0.0 with minimal standard deviations. Considering both the high silhouette coefficient and the efficient convergence rate, Hierarchical Clustering emerged as the preferable choice for clustering the Haberman dataset.

Lastly, for the CMC dataset, PCA, SMOKM, WOAKM, and GWOKM techniques showed the highest average silhouette coefficients of 0.443 ± 0.0 with minimal standard deviation. These techniques consistently produced well-separated clusters, indicating superior clustering performance. SMOKM and GWOKM demonstrated a low iteration count of 3.1 ± 0.3 with minimal standard deviations, indicating both effective clustering and computational efficiency. Therefore, these techniques provided a favorable balance of clustering performance and computational speed for the CMC dataset.

VI. Conclusions and Future Work

In this study, we extensively evaluated six well-known traditional initialization centroid techniques for K-means clustering with three proposed swarm intelligence-based techniques. The results demonstrate that the proposed swarm intelligence-based techniques surpassed all traditional centroid initialization methods for K-means clustering. This is because swarm intelligence algorithms can adapt to different dataset characteristics and adjust the number and position of centroids based on data distribution and similarity, resulting in more robust and flexible clustering results. Swarm intelligence-based techniques are also more robust to noise and outliers. Therefore, swarm intelligence-based techniques are promising for improving clustering results on various datasets, responding to the research question (RQ2). SMOKM, in particular, emerged as the superior method, consistently achieving the highest average silhouette coefficient and converging in the fewest iterations across most tested datasets. WOAKM and GWOKM proved to be on par with SMOKM, presenting themselves as equally viable alternatives for achieving commendable clustering results. These findings can respond to the research questions (RQ3 and RQ4).

The findings of this study have important implications for the field of clustering. SMOKM, with its robust performance and computational efficiency, stands out as the preferred choice for clustering diverse datasets. Its ability to produce well-defined and separated clusters makes it a valuable tool in various applications, especially when computational resources are limited.

Furthermore, other traditional techniques such as Hierarchical Clustering, PCA, and K-means++ also showcased promising results on specific datasets. These techniques can be viable alternatives to swarm intelligence-based techniques in cases where dataset characteristics or computational constraints necessitate different approaches.

As for future work, exploring the applicability and performance of swarm intelligence techniques on larger and more complex datasets would be beneficial. Additionally, investigating the combination of multiple initialization centroid techniques or the development of hybrid methods could potentially further enhance clustering performance. Furthermore, exploring the impact of parameter tuning and optimization strategies specific to each technique would be valuable to refine and maximize their effectiveness.

Overall, this research contributes valuable insights into the selection and performance evaluation of initialization centroid techniques for K-means clustering. The findings emphasize

the superiority of swarm intelligence techniques, particularly SMOKM, and guide researchers and practitioners in seeking optimal solutions for clustering diverse datasets.

Conflict of Interest

The authors have no conflict of relevant interest to this article.

References

[1] C. Yuan and H. Yang, "Research on k-value selection method of k-means clustering algorithm," *J — Multidisciplinary Scientific Journal*, vol. 2, pp. 226–235, June 2019. https://doi.org/10.3390/j2020016.

[2] T. Kodinariya and D. P. R. Makwana, "Survey on existing methods for selecting initial centroids in k-means clustering," *International Journal of Engineering Development and Research (IJEDR)*, vol. 2, no. 2, pp. 2865–2868, 2014.

[3] S. Shukla and S. Naganna, "A review on k-means data clustering approach," *International Journal of Information & Computation Technology*, vol. 4, no. 17, pp. 1847–1860, 2014.

[4] J. M. Pena, J. A. Lozano, and P. Larranaga, "An empirical comparison of four initialization methods for the k-means algorithm," *Pattern recognition letters*, vol. 20, no. 10, pp. 1027–1040, 1999. https://doi.org/10.1016/s0167-8655(99)00069-0.

[5] T. Su and J. Dy, "A deterministic method for initializing k-means clustering," in *16th IEEE international conference on tools with artificial intelligence*, (Boca Raton, FL, USA), pp. 784–786, IEEE, 15-17 November 2004. https://doi.org/10.1109/ictai.2004.7.

[6] M. E. Celebi and H. A. Kingravi, "Deterministic initialization of the k-means algorithm using hierarchical clustering," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 26, no. 07, pp. 1–25, 2012. https://doi.org/10.1142/s0218001412500188.

[7] M. E. Celebi, H. A. Kingravi, and P. A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," *Expert systems with applications*, vol. 40, no. 1, pp. 200–210, 2013. https://doi.org/10.1016/j.eswa.2012.07.021.

[8] R. T. Aldahdooh and W. Ashour, "Dimk-means" distance-based initialization method for k-means clustering algorithm"," *International Journal of Intelligent Systems and Applications*, vol. 5, no. 2, pp. 41–51, 2013. https://doi.org/10.5815/ijisa.2013.02.05.

[9] B. Li, "An experiment of k-means initialization strategies on handwritten digits dataset," *Intelligent Information Management*, vol. 10, no. 2, pp. 43–48, 2018. https://doi.org/10.4236/iim.2018.102003.

[10] V. P. Murugesan and P. Murugesan, "A new initialization and performance measure for the rough k-means clustering," *Soft Computing*, vol. 24, no. 15, pp. 11605–11619, 2020. https://doi.org/10.1007/s00500-019-04625-9.

[11] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *Electronics*, vol. 9, no. 8, pp. 1–12, 2020. https://doi.org/10.3390/electronics9081295.

[12] A. Vouros, S. Langdell, M. Croucher, and E. Vasilaki, "An empirical comparison between stochastic and deterministic centroid initialisation for k-means variations," *Machine Learning*, vol. 110, pp. 1975–2003, 2021. https://doi.org/10.1007/s10994-021-06021-7.

[13] K. Chowdhury, D. Chaudhuri, and A. K. Pal, "An entropy-based initialization method of k-means clustering on the optimal number of clusters," *Neural Computing and Applications*, vol. 33, pp. 6965–6982, 2021. https://doi.org/10.1007/s00521-020-05471-9.

[14] Z. Rahman, M. S. Hossain, M. Hasan, and A. Imteaj, "An enhanced method of initial cluster center selection for k-means algorithm," in *2021 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pp. 1–6, IEEE, 2021. https://doi.org/10.1109/asyu52992.2021.9599017.

[15] A. Torrente and J. Romo, "Initializing k-means clustering by bootstrap and data depth," *Journal of Classification*, vol. 38, no. 2, pp. 232–256, 2021. https://doi.org/10.1007/s00357-020-09372-3.

[16] S. Harris and R. C. De Amorim, "An extensive empirical comparison of k-means initialization algorithms," *IEEE Access*, vol. 10, pp. 58752–58768, 2022. https://doi.org/10.1109/access.2022.3179803.

[17] S. F. Raheem and M. Alabbas, "Optimal k-means clustering using artificial bee colony algorithm with variable food sources length.," *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 12, no. 5, pp. 5435–5443, 2022. https://doi.org/10.11591/ijece.v12i5.pp5435-5443.

[18] A. Kazemi and G. Khodabandehlouie, "A new initialisation method for k-means algorithm in the clustering problem: data analysis," *International Journal of Data Analysis Techniques and*

Strategies, vol. 10, no. 3, pp. 291–304, 2018. https://doi.org/10.1504/ijdats.2018.10015167.

[19] J. A. Alhijaj and R. S. Khudeyer, "Integration of efficientnetb0 and machine learning for fingerprint classification," *Informatica*, vol. 47, no. 5, p. 49–56, 2023. https://doi.org/10.31449/inf.v47i5.4724.

[20] G. S. Ohannesian and E. J. Harfash, "Epileptic seizures detection from eeg recordings based on a hybrid system of gaussian mixture model and random forest classifier," *Informatica*, vol. 46, no. 6, p. 105–116, 2022. https://doi.org/10.31449/inf.v46i6.4203.

[21] M. M. Mayo, "An arithmetic-based deterministic centroid initialization method for the k-means clustering algorithm," 2016.

[22] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46–61, 2014. https://doi.org/10.1016/j.advengsoft.2013.12.007.

[23] S. F. Raheem and M. Alabbas, "Dynamic artificial bee colony algorithm with hybrid initialization method," *Informatica*, vol. 45, no. 6, p. 103–114, 2021. https://doi.org/10.31449/inf.v45i6.3652.

[24] J. C. Bansal, H. Sharma, S. S. Jadon, and M. Clerc, "Spider monkey optimization algorithm for numerical optimization," *Memetic computing*, vol. 6, pp. 31–47, 2014. https://doi.org/10.1007/s12293-013-0128-0.

[25] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in engineering software*, vol. 95, pp. 51–67, 2016. https://doi.org/10.1016/j.advengsoft.2016.01.008.

[26] A. Q. Obaid and M. Alabbas, "Hybrid variable-length spider monkey optimization with good-point set initialization for data clustering," *Informatica*, vol. 47, no. 8, p. 67–78, 2023. https://doi.org/10.31449/inf.v47i8.4872.