Hybrid Variable-Length Spider Monkey Optimization with Good-Point Set Initialization for Data Clustering

Athraa Qays Obaid, Maytham Alabbas*
Department of Computer Science, College of Computer Science and Information Technology, University of Basrah, Basrah, Iraq
E-mail: itpg.athraa.qays@uobasrah.edu.iq, ma@uobasrah.edu.iq
*Corresponding author

Keywords: data clustering, spider monkey optimization, k-means clustering, variable-length spider monkey optimization, good-point set

Received: May 21, 2023

Data clustering refers to the process of grouping similar data points based on patterns or characteristics. It finds applications in image analysis, pattern recognition, and data mining. The k-means algorithm is commonly used for this purpose, but it has two main limitations. Firstly, it necessitates the user to explicitly specify the number of clusters. Secondly, it is highly sensitive to the initial selection of cluster centroids. To overcome these limitations, this study presents a novel approach that utilizes a variable-length spider monkey optimization algorithm (VLSMO) with a hybrid measure to determine the optimal number of clusters and initial centroids. Experimental results obtained from real-life datasets demonstrate that VLSMO outperforms the standard k-means algorithm and other techniques in terms of accuracy and clustering capacity.

Povzetek: Prispevek opisuje novo metodo za združevanje podatkov z uporabo optimirnega algoritma VLSMO, ki odpravlja omejitve predhodnih algoritma, izboljšuje natančnost in zmogljivost združevanja.

1 Introduction

Data clustering is one of the most important data mining approaches, which involves partitioning data instances into smaller groups, where each group comprises objects that are similar to each other but distinct from those in other clusters. Clusters are defined by a center point and a proximity metric that measures the similarity or dissimilarity of the candidate data points. Clustering analysis has as its primary objective the creation of clusters consisting of the highest density of similar points and the most distant clusters of different points. Clustering cannot be performed manually due to the large volume of data. Instead, specialized computing techniques are used. Nonetheless, clustering differs from classification since most data is unlabeled, implicitly performing classification. Therefore, it is considered unsupervised learning.

The practice of clustering is prevalent throughout industries, as it allows related objects to be grouped. For example, a clustering approach can be used in marketing to identify consumers with similar purchasing habits. In educational settings, it can be useful in analyzing students' academic achievement by grouping those with similar study habits. In addition, clustering can also be utilized in diverse applications, including the segmentation of images, the detection of outliers, the detection of tumors, and the detection of fraud. Furthermore, clustering is a powerful method of uncovering hidden patterns within a dataset. Despite its wide range of applications, clustering poses a number of challenges [1].

Many clustering algorithms have been developed, including K-means, density-based spatial clustering of applications with noise (DBSCAN), expectation maximization (EM), and hierarchical agglomerative clustering (HAC). The K-means algorithm is widely recognized as the most commonly employed clustering algorithm. The K-means algorithm is popular in scientific research and industrial applications because it is simple, fast convergence, and scalable. Nevertheless, k-means clustering, which involves randomly distributing starting points during center initialization, frequently results in local optimal clustering outcomes that could result in inaccurate categorization due to instability. It has a number of limitations, including the necessity of specifying the number of clusters and its sensitivity to initial center points. In order to overcome the limitations of this algorithm, a globally optimized approach must be adopted [2]. Several techniques have been proposed in the literature to overcome these limitations, including the elbow method, the gap statistic, and the canopy method. The appropriate number of clusters (k) can be determined using these techniques. Furthermore, various algorithms can be used to identify the initial centroids, including the Forgy method, random partition method, and k-means++ algorithm. As far as the authors are aware, there are limited techniques available to determine both parameters simultaneously based on using optimization techniques like genetic algorithms (GA), artificial bee colony (ABC), and particle swarm optimization (PSO).

The current work aims to progress in this area by presenting a modified version of the spider monkey optimization algorithm (SMO) with variable length to determine the number of clusters and their initial centroids.

The following sections of this paper are organized as follows: An overview of related work is presented in Section 2. Section 3 provides a theoretical background on the topic. Section 4 describes the current work. A summary of the results is presented in Section 5. Finally, the paper is concluded in Section 6.

2 Literature review

Numerous techniques have been presented to determine suitable parameters for the k-means algorithm. Several of these techniques focus on estimating the number of clusters (k), others optimize the initial centroids, and some address both aspects simultaneously.

Multiple methods are available for estimating the number of clusters, and selecting a suitable one depends on the dataset and goals. One of the most common methods is the elbow method. It works by plotting the sum of squared distances between each data point and its nearest centroid against different values of k. The point where the curve starts to level off is considered the optimal value of k [3]. An alternative approach is the silhouette method, which calculates the silhouette coefficient for each point in order to assess how well it fits within the assigned cluster in relation to other clusters. The optimal number of clusters is determined by selecting the k value that maximizes the average silhouette coefficient. Additionally, the gap statistic provides a robust method in this regard. In this method, the within-cluster sum of squares of observed data is compared to a randomly generated data set to identify the optimal value of k that minimizes the gap statistics. Similarly, the canopy method uses hierarchical clustering to group data points. It determines the optimal number of clusters by identifying the level in the hierarchy at which clusters are most distinctive. These are just a few of the many methods that can be used to estimate the number of clusters.

On the other hand, choosing suitable initial centroids is also crucial as they serve as the starting points for the k-means clustering algorithm and greatly influence the final clustering outcomes. The most common method of selecting initial centroids is to choose them randomly from the dataset. Alternatively, more advanced algorithms such as k-means++ can be used to refine the selection process. It aims to select centroids that are distant from each other and representative of the data distribution. This helps to avoid local optima, which can occur when the initial centroids are not well-chosen. The authors in [4] propose a hybrid algorithm, ACO-K-means, that combines the ant colony optimization (ACO) algorithm with the k-means algorithm for clustering to find good initial centroids for the k-means algorithm, which can improve the clustering results. ACO-K-means outperform the k-means algorithm on the tested datasets. A hybrid clustering algorithm named MABCKM is presented in [5], which combines modified ABC and KM algorithms. MABCKM addresses the issue of dependency on initial cluster centers in KM. A cooperative algorithm is presented in [6] that combines PSO with k-means to provide a global and local search capability. Experimental results indicate that the proposed algorithm achieves satisfactory efficiency and robustness when clustering data. In [7], three K-means initialization strategies are compared using the UCI ML handwritten digits dataset: random, K-means++, and PCA-based K-means. The findings indicate that the PCA-based K-means initialization strategy performs better than the other two approaches regarding accuracy and running time. In [8], the authors prove that combining the glowworm swarm optimization (GSO) algorithm, k-means algorithm, and good-point set can improve the clustering effect and stability under unsupervised learning conditions.

Further, some studies have investigated using meta-heuristic algorithms to optimize k-means issues. These algorithms search the parameter space to find the optimal combination of k and initial centroids. For instance, an algorithm based on variable-length chromosomes is proposed in [9] for solving the K-means clustering problem. This approach is designed to automatically determine the number of cluster centers that is most appropriate. In [10], the authors propose a new method KMBA for optimizing the k-means clustering algorithm using the bat algorithm. In KMBA, each bat in the algorithm performs two essential tasks: (1) identify the ideal number of clusters utilizing a discrete PSO approach based on CPSO, and (2) determine the optimal number of cluster centers using the K-means algorithm. It is easy to implement and effective in a variety of problems. In [11], a new K-FA approach combines the firefly algorithm (FA) with k-means clustering to cluster data. It utilizes FA to determine the centroids of the specified number of clusters and further refines them using k-means clustering. K-FA algorithm clusters data accurately and effectively. In [2], the authors present a novel approach, ABCVL (Artificial Bee Colony with Variable-Length Individuals), for optimizing the k-means clustering algorithm. ABCVL is used to determine both the value of k and the initial centroids. ABCVL has been evaluated on various datasets, demonstrating that it is more accurate in clustering than the standard k-means algorithm. ABCVL is also simple to implement and is effective across a wide range of problem domains.

Finally, the literature indicates that there is no one-size-fits-all approach to estimating k-means parameters. It is important to consider both the data characteristics and the problem at hand before choosing a method. Therefore, it is essential to experiment with different techniques and compare their performance to select the most appropriate one for a particular application.

Table 1 summarizes a selection of relevant studies in the field.

Table 1: Relevant studies in the field.

Ref	**Year**	**Technique**	**fitness**	dataset	**Best**	**Mean**	**SD**	**Run**
[3]	2019	Analyze four K-value selection algorithms Elbow Method, Gap Statistic, Silhouette Coefficient, and Canopy	Silhouette Coefficient	▪ Iris	0.765	---	---	---
[4]	2005	Hybrid Ant Colony Optimization with K-Means Algorithm (ACO-K-means algorithm)	distance between cluster centers and calculating the sum of color and physical distances between each pixel and its assigned cluster center.	▪ images	---	---	---	---
[5]	2011	Hybrid Modified Artificial Bee Colony and K-Means Algorithm (MABCKM)	Euclidean distance (short for distance)	▪ Iris ▪ glass ▪ lung cancer ▪ soybean (small) ▪ wine ▪ vowel	78.8514 336.0840 535.6552 208.1545 2.3707e6 3.06907	78.8516 336.6181 570.8732 227.6273 2.3708e6 3.06907	0.00004 0.1604 10.5690 10.2121 29.9780 10.1211	20
[6]	2012	A cooperative algorithm based on PSO and k-means	Sum of Intra Cluster Distances (SICD)	▪ Irish ▪ Pima ▪ Wine ▪ WDBC ▪ Glass ▪ Sonar	96.73000 47561.35 16292.68 211.0400 149473.1 234.6300	96.91000 47580.43 16293.75 214.8300 149475.0 234.7800	0.170000 59.97000 0.880000 4.090000 1.370000 0.160000	120
[7]	2018	Evaluation of these three strategies for initializing the centroids	silhouette coefficient	▪ digits	0.15	---	---	---
[8]	2018	Glowworm Swarm Optimization, K-means, and Good-Point Set (GSOK-GP algorithm)	Density-Based	▪ Iris ▪ Glass	97.32 225.08	89.33 53.50	0 0.0090	20
[9]	2019	genetic algorithm with variable length string	Davies-Bouldin Index (DBI)	▪ Random	0.2126	---	---	20
[10]	2012	K-Means and Bat Algorithm (KMBA)	Accuracy	▪ Iris	95%	---	---	---
[11]	2012	hybrid K-means and Firefly Algorithm (K-FA)	intra-cluster distance	▪ Iris ▪ WDBC ▪ Sonar ▪ Glass ▪ Wine	96.13 149450.3 229.35 210.51 16284.01	103.87 149590.2 231.36 221.87 16327.53	2.45 197.31 2.94 10.5 10.10	30
[2]	2022	Improved artificial bee colony algorithm with variable-length individuals (ABCVL)	Hybrid-Scale2	▪ Mall Customers ▪ Digits ▪ Breast Cancer	0.521397 1.336862 0.423736	0.520917 1.345410 0.423736	0.002586 0.008708 1.110×10^{-16}	30

3 Background

3.1 Spider monkey optimization (SMO) algorithm

SMO is a collaborative, iterative process based on trial and error, much like the other population-based algorithms. Six phases comprise the SMO process: Local Leader, Global Leader, Local Leader Learning, Global Leader Learning, Local Leader Decision, and Global Leader Decision. The Gbest-guided ABC [12] [13] and a modified version of ABC [14] [15] inspired the position updating procedure in the Global Leader phase. Following is an explanation of each SMO implementation step in detail
[16]

3.1.1 Initialization of the population

Initially, SMO creates an initial population of N spider monkeys (SMs) with a uniform distribution. Each SM in the population is represented by a D-dimensional vector, SM_{ij} (i = 1, 2,..., N and j = 1, …, D), where D represents the length of the optimization problem. Each SM represents a possible solution to the issue. Every SM_i is initialized as follows:

$$SM_{ij} = SM_{minj} + U\,(0,1) \times (SM_{maxj} - SM_{minj}). \qquad (1)$$

Here, U (0,1) is a uniformly distributed random number in the range [0,1], and SM_{minj} and SM_{maxj} are the limits of SM_i in the jth direction.

3.1.2 Local leader phase (LLP)

Each SM adjusts its present position during LLP based on information from the experience of the local leader and local group members. The fitness value of the newly acquired position is computed. As long as the new position has a greater fitness value than the old one, the SM replaces the old one with the new one. In LLP, the position update equation for the ith SM, a member of the kth local group, is:

$$SM_{newij} = SM_{ij} + U(0,1) \times (LL_{kj} - SM_{ij}) + U(-1,1) \times (SM_{rj} - SM_{ij}), \quad (2)$$

where, LL_{kj} stands for the jth dimension of the kth local group leader position, and SM_{ij} is the jth dimension of the ith SM. In order to ensure that r≠i and U (0,1) is a uniformly distributed random integers between 0 and 1, SM_{rj} is the jth dimension of the rth SM that is randomly selected inside the kth group. Algorithm 1 illustrates the method of updating LLP positions [16].

Algorithm 1: LLP

MG: the swarm's maximum number of groups
Pr: the perturbation rate ∈ [0.1, 0.9], determining how much the current location is perturbed.

```
for each k ∈ {1, ..., MG} do
   for each member $SM_i$ ∈ kth group do
     for each j ∈ {1, ..., D} do
       if  U (0,1) ≥ pr then
            Update $SM_{newij}$ using Eq. (2).
       else
            $SMnew_{ij} = SM_{ij}$
       end if
     end for
   end for
end for
```

3.1.3 Global leader phase (GLP)

Once LLP has been completed, GLP will begin. Each SM updates its position during the GLP based on the experience of the global leader and the local group members. Here, the position is updated by applying Eq. (3).

$$SM_{newij} = SM_{ij} + U(0,1) \times (GL_j - SM_{ij}) + U(-1,1) \times (SM_{rj} - SM_{ij}). \quad (3)$$

Where j ∈ {1, 2, ..., D} is the randomly selected index, and GL_j is the jth dimension of the global leader position.

In this phase, SM placements are adjusted based on probabilities derived from their fitness. A better candidate has a higher chance of improving under this approach. For determining the probability $prob_i$, Eq. (4), or some other equation related to fitness, may be used:

$$prob_i = 0.9 \times \frac{fitnessi}{max\_fitness} + 0.1, \quad (4)$$

where, $fitness_i$ denotes the ith SM's fitness value and max_ fitness denotes the group's maximum fitness. Additionally, the fitness of the SMs' newly produced position is computed, compared to the previous one, and the more suitable position is accepted. Algorithm 2 illustrates the process of updating GLP positions [16].

Algorithm 2: GLP

```
for k = 1 to MG do
   count = 1;
     GS = kth group size;
     while count < GS do
       for i = 1 to GS do
         if  U (0, 1) < $prob_i$ then // see Eq. (4)
             count = count + 1.
             Randomly select j ∈ {1...D}.
             Randomly select $SM_r$ from kth group s.t. r ≠ i.
             Update $SM_{newij}$ using Eq. (3).
         end if
       end for
       if i is equal to GS then
          i = 1;
       end if
  end while
end for
```

3.1.4 Global leader learning (GLL)

During this phase, the position of the global leader is updated by applying greedy selection on the population, i.e., selecting the SM position with the best fitness among the population as the updated position. In addition, the global leader position is determined, and if it is not updated, the GlobalLimitCount is incremented by one.

3.1.5 Local leader learning (LLL) phase

In this phase, the local leader's position is updated using a greedy selection method within the group. The local leader is chosen based on the SM's fitness level. The updated position of the local leader is then compared to its previous position, and if the local leader has not changed, the LocalLimitCount is incremented by one.

3.1.6 Local leader decision (LLD) phase

Assume that the position of any local leader is not updated until the Local Leader Limit has been reached. In this situation, all group members update their positions using either random initialization or the combined knowledge of the Global Leader and Local Leader obtained by Eq. (5), depending upon the pr.

$$SM_{newij} = SMij + U(0,1) \times (GL_j - SM_{ij}) + U(0, 1) \times (SM_{ij} - LL_{kj}) \quad (5)$$

According to Eq. (5), the updated dimension of this SM is attracted to the global leader and repellent to the local leader. Algorithm 3 displays the process of the LLD phase, where LocalLimitCountk refers to the trial counter for the local best solution of the kth group.

Algorithm 3: LLD Phase

```
for k = {1...MG} do
  if LocalLimitCountk > Local Leader Limit then
    LocalLimitCount$_k$ = 0.
    GS = kth group size;
    for i ∈ {1...GS} do
      for each j ∈ {1...D} do
        if U (0,1) ≥ pr then
          $SM_{newij} = SM_{minj} + U(0,1) \times (SM_{maxj} - SM_{minj})$
        else
          Upate $SM_{newij}$ using Eq. (5).
        end if
      end for
    end for
  end if
end for
```

3.1.8 Global leader decision (GLD) phase

As soon as the position of the global leader has not been changed for a predefined number of iterations, the population is divided into smaller groups by the global leader. The population is divided into two groups first, followed by three groups, and so on until the maximum number of groups (MG) has been reached. Whenever a new group is established during the GLD phase, the LLL procedure is initiated to elect the local leader. Regardless of how many groups are established, if the global leader's position remains unchanged, all groups will be united. Hence, this method was based on SMs' fusion-fission structure. Algorithm 4 illustrates how the GLD phase works.

Algorithm 4: GLD Phase

```
if GlobalLimitCount > Global Leader Limit then
  GlobalLimitCount = 0
  if Number of groups < MG then
    Divide the population into groups.
  else
    Combine all the groups to make a single group.
  end if
  Update Local Leaders position.
end if
```

The SMO algorithm can be summarized in Algorithm 5 [16].

Algorithm 5: SMO Algorithm

```
1. Initialize the population of spider monkeys.
2. Evaluate the fitness of each spider monkey.
3. Identify the best spider monkey based on fitness.
4. Perform exploration and exploitation by updating the positions of
   spider monkeys.
5. Evaluate the fitness of the updated spider monkeys.
6. If the termination condition is met, stop; otherwise, go to step 3.
7. Output the best solution found.
```

3.2 K-Means clustering algorithm

The k-means algorithm and its extensions have a long history, and it remains a challenging task today. This is because it depends on the initializations and requires determining a specific number of clusters [17].

This algorithm partitions the dataset into k unique clusters by iteratively achieving a local minimum. The initial K cluster centers for the K-means method are selected at random from the dataset, where the user predetermines k. Throughout each iteration, each point in a dataset is assigned to the nearest cluster center. Once all data points have been grouped into clusters, the new centroid is calculated as the mean of all cluster points for each cluster. This process is repeated until either the centroids of the clusters do not change any longer or the maximum number of iterations has been reached [2] [18].

3.3 Basic theory of good-point set

Good point sets are defined and structured in the following manner [8] [19]:

1) Assume G_s is a unit cube in S-dimensional Euclidean space, which is expressed as $x \in G_s$,

$$x = (x_1, x_2,...,x_s)\ , \tag{6}$$

where, $0 \le X_i \le 1$, $i = 1, 2,..., s$.

2) Assume $P_n(K)$ is a point set with the number of n in G_s, which is expressed as:

$$P_n\,(k) = \{(X_1^{(n)}(K), X_2^{(n)}(K), ..., X_s^{(n)}(K))\}\ , \tag{7}$$

where, $0 \le K \le n$, $0 \le x_i^{(n)}\,(K) \le 1$, $i = 1, 2,..., s$.

3) Assume $r = (r1, r2, ..., r_s)$ is a given point in G_s and $N_n(r) = N_n(r1, r2, ..., r_s)$ is the number of points not satisfying the inequality below in point set $P_n(k)$.

$$0 \le X_i^{(n)}\,(k) \le r_i,\ \text{where, } i = 1, 2, ..., s. \tag{8}$$

$\varphi(n) = \sup |N_n(r)/n - |r||$, where $r \in G_s$, $|r| = r_1 r_2 \cdots r_s$, and $\varphi(n)$ is the deviation of point set $P_n(K)$.

4) Assume $\varphi(n)$ is the deviation of $P_n(K) = \{(X_1^{(n)}{*}k, X_2^{(n)}{*}K,..., X_s^{(n)}{*}K,\ K= 1, 2, ..., n\}$ and meets the requirements below:

$\varphi(n) = C(r,\varepsilon)n^{-1+\varepsilon}$, where $C(r, \varepsilon)$ is a constant related to r and ε, $\varepsilon > 0$.

A good-point set is denoted by $P_n(K)$, and a good point is denoted by r. The order of deviation $\varphi(n)$ about approximation integration is shown by applicable theorems to be dependent solely on n and independent of the sample's spatial dimensions. The computation in high-dimensional spaces may thus be supported more effectively by the good-point set. Meanwhile, the deviation $\varphi(n)$ of n points $P_n = X_1, X_2, ..., X_n$ acquired by a good-point set is much better than n points produced by a random technique for a point set object whose distribution is unknown. Based on this characteristic of the good-point set, a better initial distribution strategy may be offered for the swarm distribution in the swarm intelligence algorithm.

4 The current work

Swarm intelligence (SI) is significantly limited when dealing with fixed-length individuals. This paper proposes an enhanced version of the SMO algorithm, known as the variable-length SMO algorithm (VLSMO), that handles variable length. To check the effectiveness of VLSMO, we will utilize it to tackle the challenges of the k-means algorithm by determining the number of clusters (k), and the initial cluster centers.

We will describe the current work in the following lines.

4.1 Individual representation

Two representation techniques are used to represent individuals (SMs) in the population: indirect and direct. Suppose the dataset contains 100 data points numbered from 1 to 100, each data point has four attributes, and the number of clusters is 3. The following is a description of SM with the details below using these two techniques.

Data point index	Attributes (4 features)			
2	4.7	3.2	1.3	0.2
40	5.0	3.5	1.3	0.3
70	6.4	2.9	4.3	1.3

- Indirect technique: in this technique, every SM position is represented as the index of data points in the dataset, where each index corresponds to a sample. For example, below is the representation of SM according to this technique.

Cluster 1	Cluster 2	Cluster 3
2	40	74

This individual implies that the number of clusters is 3, and the first centroid corresponds to the data point with index 2, the second centroid corresponds to the data point with index 40, and the third centroid corresponds to the data point with index 74.

- Direct technique: in this technique, every SM position is represented as the attributes of data points in the dataset, where each position corresponds to a sample. For example, below is the representation of SM according to this technique.

4.7	3.2	1.3	0.2	Cluster 1
5.0	3.5	1.3	0.3	Cluster 2
6.4	2.9	4.3	1.3	Cluster 3

This individual implies that the number of clusters is 3. The first centroid corresponds to the data point with attributes (4.7, 3.2, 1.3, 0.2) instead of its index 2; the second centroid corresponds to the data point with attributes (5.0, 3.5, 1.3, 0.3) instead of its index 40; and the third centroid corresponds to the data point with attributes (6.4, 2.9, 4.3, 1.3) instead of its index 74.

4.2 Initial population based on the Good-point set

To optimize the SMO, it is effective to switch from a random distribution to a uniform distribution in the search space. Nevertheless, a good-point set method must be applied to enhance the initial distribution to ensure that the solution space is covered in most cases. This can be accomplished through the theory of the good-point set, discussed in Section 3.3.

The population in this context exhibits the following attributes: it comprises individuals {ind_1, ind_2, ind_3, …, N}, and the length of each individual corresponds to the number of clusters, denoted as K, where $nC_{min} \leq K_i \leq nC_{max}$ for $1 \leq i \leq N$. Here, nC_{min} and nC_{max} denote the minimum and maximum number of clusters, respectively. Each individual serves as a representation of the initial cluster centroids (C): $ind_i = \{C_1, C_2, C_3, \dots C_{Ki}\}$.

4.3 Objective function

We evaluated the clustering quality by experimenting with eight objective functions that relied on key metrics. These metrics encompassed both internal (1-2) and external (3-6) measures, as well as Inertia (7) and a hybrid measure we proposed (8). These measurements are explained below [1].

1. **Davies-Bouldin Index (DBI) measure** measures the average similarity between a cluster and its most comparable one. The lower the DBI value, the better the clustering, while zero indicates optimal clustering.
2. **Silhouette coefficient measure:** It measures how similar a data point is to its own cluster compared to others. The score ranges from -1 to 1, where a higher value indicates better clustering results.
3. **Homogeneity measure:** This measure assesses the degree to which all data points within a cluster belong to the same class. A high homogeneity score indicates well-separated clusters.
4. **Completeness measure:** It measures how effectively all data points belonging to a single class are placed together. A high completeness score indicates optimal clustering.
5. **V-measure**: It assesses both homogeneity and completeness at the same time. It measures how well the clustering solution matches the known class labels. Better clustering is indicated by higher V-measure scores, with 1 being perfect clustering.
6. **Adjusted rand index (ARI) measure**: It measures the similarity between predicted and actual clustering by considering the possibility of chance clustering. The score ranges from -1 to 1, where a higher value indicates better clustering results.
7. **Inertia:** It measures the sum of squared distances between each data point and the center of the cluster to which it belongs. The lower the score, the better the result [2].
8. **Hybrid score (HS):** this proposed measure integrates the Silhouette and ARI measures, assigning a weight

to each measure to determine its relative importance using Eq. (9).

$$HS = a \times \text{Silhouette} + b \times \text{ARI}, \qquad (9)$$

where a and b are real numbers within the range [0,1] and a+b =1.

4.4 Modified search equation

The SMO algorithm was improved to accommodate individuals of varying lengths. By adhering to numerous modification instructions, The LLP equation has been revised to accommodate two scenarios based on the potential varying lengths of the two individuals, namely MLLPq. Assume we have two individuals (Ind_1 and Ind_2) of different lengths (L_1 and L_2). Detailed descriptions of each scenario are provided below.

- **MLLPq$_1$**

In the first scenario, MLLPq$_{1,}$ the L_1 is less than or equal to the L_2. Equation (2) is utilized on the initial L_1 positions shared by the two individuals (Ind_1 and Ind_2), resulting in a new individual (Ind_3) with a length equal to L_1. The sketch diagram of the MLLPq$_1$ example is shown in Figure 1(a).

- **MLLPq$_2$**

The second scenario, MLLPq$_{2,}$ involves L_1 being greater than L_2. Initially, Eq. (2) is utilized on the initial L_2 positions shared by the two individuals (Ind_1 and Ind_2), resulting in the first L_2 positions in the new individual (Ind_3). Then, a random integer number P in range (L_2, L_1] is selected. Finally, the positions from L_2+1 to P of Ind_1 are copied to the same positions at Ind_3. In this scenario, the length of Ind_3 ranges from L_2 to L_1. The sketch diagram of the MLLPq$_2$ example is shown in Figure 1(b).

Figure 1 illustrates how a new individual Ind_3 can be created by combining two individuals, Ind_1 and Ind_2, with lengths L_1 and L_2, respectively.

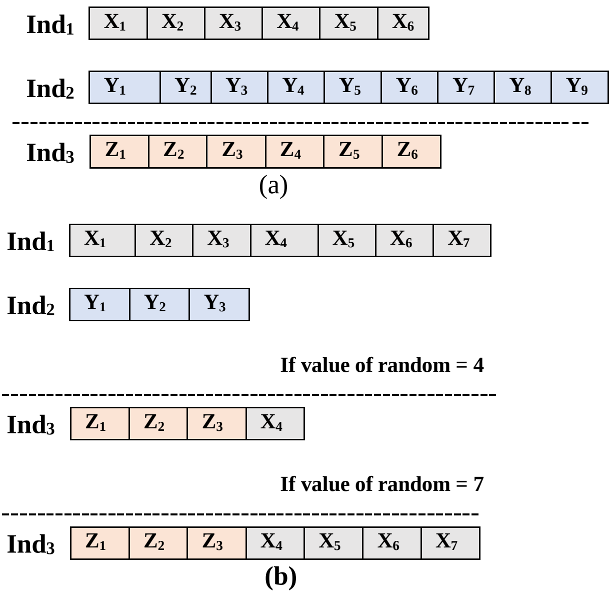


Figure 1: The schematic diagram of (a) MLLPq$_1$, and (b) MLLPq$_2$.

5 Results

5.1 Tested datasets

The current work evaluates five datasets from the UCI repository [20], and their properties of these datasets are listed in Table 2.

Table 2: Characteristics of the data sets considered

Name	No. of classes	No. of attribute	Size of dataset
Seed	3	7	210 (70, 70, 70)
Bupa	2	6	345 (145, 200)
Iris	3	4	150 (50, 50, 50)
Heart	2	13	270 (120, 150)
Haberman	2	3	306 (225, 81)

5.2 Parameters setting

The current work employs the following parameters settings for all experiments:

- The swarm size: N = 20
- The minimum number: $nC_{min} = 2$
- The maximum number nCmax = 12
- The GlobalLeaderLimit = [N/2, 2 × N]
- The LocalLeaderLimit = D × N
- The maximum number of iterations = 100
- a and b for HS function = 0.5

5.3 Results

We conducted a series of experiments to assess the effectiveness of the VLSMO method from different perspectives. These experiments are referred to as Experiment I, Experiment II, and Experiment III.

In Experiment I, our objective was to evaluate the impact of the performance of the objective function on the overall effectiveness of the VLSMO method. We carefully examined how variations in the objective function affected the performance and optimization capabilities of the VLSMO method.

Moving on to Experiment II, our focus was on analyzing the efficiency of the VLSMO method by considering individuals of varying lengths. We investigated how the method's performance was influenced by different lengths of individuals, aiming to identify any correlations between the length of individuals and the efficiency of the VLSMO method.

In Experiment III, we aimed to assess the efficiency of the VLSMO method by comparing it with other techniques that have been previously discussed in [23]. This experiment provided valuable insights into how the VLSMO method compared to other existing techniques, allowing us to better understand its strengths and weaknesses.

Through these experiments, we comprehensively understood the VLSMO method and its performance across different dimensions.

5.3.1 Experiment I

In this experiment, we evaluated the effectiveness of the VLSMO method for clustering problems on five datasets, as described in Section 5.1, using eight measures defined in Section 3.4.3 as objective functions. Tables 3-13 summarize the results of this experiment, including the best value, mean, optimal k, and mean of k, with the best outcomes highlighted in bold typeface.

Table 3: Experiment I on Seed dataset-Indirect technique.

Measure	Best	Mean	Optimal (K)	Mean (K)
DBI	0.667283	0.667283	2	2.0
Silhouette	0.474919	0.485367	2	2.2
Homogeneity	0.133885	0.133702	9	10.0
Completeness	0.256622	0.256622	2	2.0
V-measure	0.282329	0.282353	3	3.0
Inertia	161.2699	162.6081	12	12.0
ARI	0.27389	0.277021	3	3.1
HS	0.400861	0.400861	**3**	3.0

Table 4: Experiment I on Bupa dataset-Indirect technique.

Measure	Best	Mean	Optimal (K)	Mean (K)
DBI	0.767878	0.767878	2	2.0
Silhouette	0.365595	0.384366	2	2.3
Homogeneity	0.868698	0.878894	11	10.6
Completeness	0.961773	0.966154	10	9.6
V-measure	0.938959	0.943134	11	11.1
Inertia	113115.8	115453.6	12	11.9
ARI	0.961553	0.968388	6	8.3
HS	0.685475	0.68636	**2**	2.1

Table 5: Experiment I on Iris dataset-Indirect technique.

Measure	Best	Mean	Optimal (K)	Mean (K)
DBI	0.404293	0.455829	2	2.2
Silhouette	0.318954	0.344599	2	2.2
Homogeneity	0.022699	0.031922	12	10.9
Completeness	0.116486	0.146221	2	2.3
V-measure	0.219169	0.219169	3	3.0
Inertia	22.39424	22.66280	12	11.9
ARI	0.213836	0.224934	3	3.1
HS	0.336311	0.34281	**3**	3.3

Table 6: Experiment I on Heart dataset-Indirect technique.

Measure	Best	Mean	Optimal (K)	Mean (K)
DBI	0.713265	0.727587	3	3.3
Silhouette	0.614592	0.618281	2	2.3
Homogeneity	0.783788	0.791029	11	11.1
Completeness	0.927299	0.932774	6	6.7
V-measure	0.89814	0.902304	8	9.4
Inertia	154717.8	164091.2	12	11.3
ARI	0.904502	0.909118	3	3.8
HS	0.793613	0.794022	**2**	**2.2**

Table 7: Experiment I on Haberman dataset-Indirect technique

Measure	Best	Mean	Optimal (K)	Mean (K)
DBI	0.80366	0.800347	3	4.2
Silhouette	0.566715	0.567513	3	3.0
Homogeneity	0.834284	0.845633	11	10.4
Completeness	0.904877	0.910012	2	2.1
V-measure	0.914638	0.916857	2	2.1
Inertia	6472.869	6552.305	12	11.9
ARI	0.811333	0.82498	2	2.2
HS	0.734413	0.739459	**2**	2.5

Table 8: Experiment I on Seed dataset-direct technique.

Measure	Best	Mean	Optimal (K)	Mean (K)
DBI	0.667282	0.667282	2	2.0
Silhouette	0.474919	0.490591	2	2.3
Homogeneity	0.135167	0.147534	11	11.1
Completeness	0.256622	0.271856	2	2.4
V-measure	0.282328	0.282376	3	3.0
Inertia	161.591242	166.966275	12	11.7
ARI	0.2738904	0.277063	3	3.1
HS	0.400861	0.400861	**3**	3.0

Table 9: Experiment I on Bupa dataset-direct technique.

Measure	Best	Mean	Optimal (K)	Mean (K)
DBI	0.767878	0.739907	2	3.8
Silhouette	0.365594	0.369024	2	2.2
Homogeneity	0.893468	0.901519	11	11.1
Completeness	0.968224	0.968492	9	10.2
V-measure	0.950264	0.953214	11	10.1
Inertia	113166.23	114804.18	12	11.8
ARI	0.969933	0.972824	6	7.1
HS	0.685474	0.687244	**2**	2.2

Table 10: Experiment I on Iris dataset-direct technique.

Measure	Best	Mean	Optimal (K)	Mean (K)
DBI	0.404292	0.404292	2	2.0
Silhouette	0.318953	0.331776	2	2.1
Homogeneity	0.032809	0.041085	8	9.0
Completeness	0.116485	0.155215	2	2.4
V-measure	0.219168	0.2202207	3	3.2
Inertia	22.507321	23.250523	12	11.7
ARI	0.1984501	0.21826009	3	3.1
HS	0.3363106	0.342809	**3**	3.0

Table 11: Experiment I on Heart dataset-direct technique.

Measure	Best	Mean	Optimal (K)	Mean (K)
DBI	0.713264	0.738831	3	3.2
Silhouette	0.614592	0.615822	2	2.1
Homogeneity	0.798409	0.800753	12	11.5
Completeness	0.929894	0.9351805	6	6.8
V-measure	0.900645	0.905123	10	9.7
Inertia	155568.13	163630.64	12	11.4
ARI	0.895478	0.9088609	3	3.7
HS	0.793613	0.793817	**2**	2.1

Table 12: Experiment I on Haberman dataset-direct technique

Measure	Best	Mean	Optimal (K)	Mean (K)
DBI	0.727124	0.740409	7	6.6
Silhouette	0.566714	0.567868	3	3.1
Homogeneity	0.835039	0.846492	12	10.4
Completeness	0.893457	0.905006	2	2.3

V-measure	0.914637	0.919636	2	2.0
Inertia	6466.24	6521.739	12	12.0
ARI	0.799551	0.822034	2	2.4
HS	0.727788	0.734275	**2**	2.1

Table 13: Accuracy of the tested measures on the five datasets-Indirect and direct technique.

Measure	Seed	Bupa	Iris	Heart	Haberman	Accuracy
DBI	✕	✔	✕	✕	✕	10%
Silhouette	✕	✔	✕	✔	✕	20%
Homogeneity	✕	✕	✕	✕	✕	0%
Completeness	✕	✕	✕	✕	✔	10%
V-measure	✔	✕	✔	✕	✔	30%
Inertia	✕	✕	✕	✕	✕	0%
ARI	✔	✕	✔	✕	✔	30%
HS	✔	✔	✔	✔	✔	**100%**

5.3.2 Experiment II

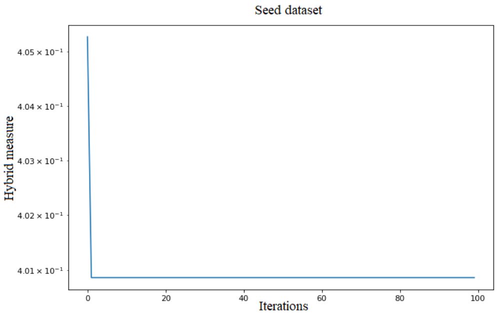
In this experiment, we evaluated the effectiveness of the VLSMO method for optimizing k-means techniques across five clustering datasets. We contrasted the outcomes against those of the standard k-means algorithm. Tables 14-15 summarize the outcomes of this experiment, displaying the best value, mean, and standard deviation (SD) generated by each algorithm throughout 30 runs, with the best results highlighted in bold typeface. Furthermore, Figures 2 and 3 depict the evolution curve of the VLSMO algorithm across the five clustering datasets.

Table 14: The results-Indirect technique of Experiment II.

Dataset	Basic K-Means Algorithm			VLSMO Algorithm		
	Best	Mean	SD	Best	Mean	SD
Seed	0.4052703	0.4052703	1.110223	**0.400861**	0.400861	2.775557
Bupa	0.687704	0.687704	0.0	**0.685475**	0.690008	0.014974
Iris	0.358471	0.358471	1.665334	**0.336311**	0.34281	0.002166
Heart	0.797463	0.797463	2.220446	**0.793613**	0.793886	0.000694
Haberman	0.800257	0.800257	1.110223	**0.731062**	0.736556	0.005237

Table 15: The results-direct technique of Experiment II.

Dataset	Basic K-Means Algorithm			VLSMO Algorithm		
	Best	Mean	SD	Best	Mean	SD
Seed	0.405270	0.405270	1.11022	**0.400861**	0.401992	0.006093
Bupa	0.687704	0.687704	0.0	**0.685474**	0.688840	0.013460
Iris	0.358471	0.358471	1.66533	**0.336310**	0.346141	0.010807
Heart	0.797463	0.797463	2.22044	**0.793613**	0.793954	0.000761
Haberman	0.800257	0.800257	1.11022	**0.727788**	0.735944	0.006259



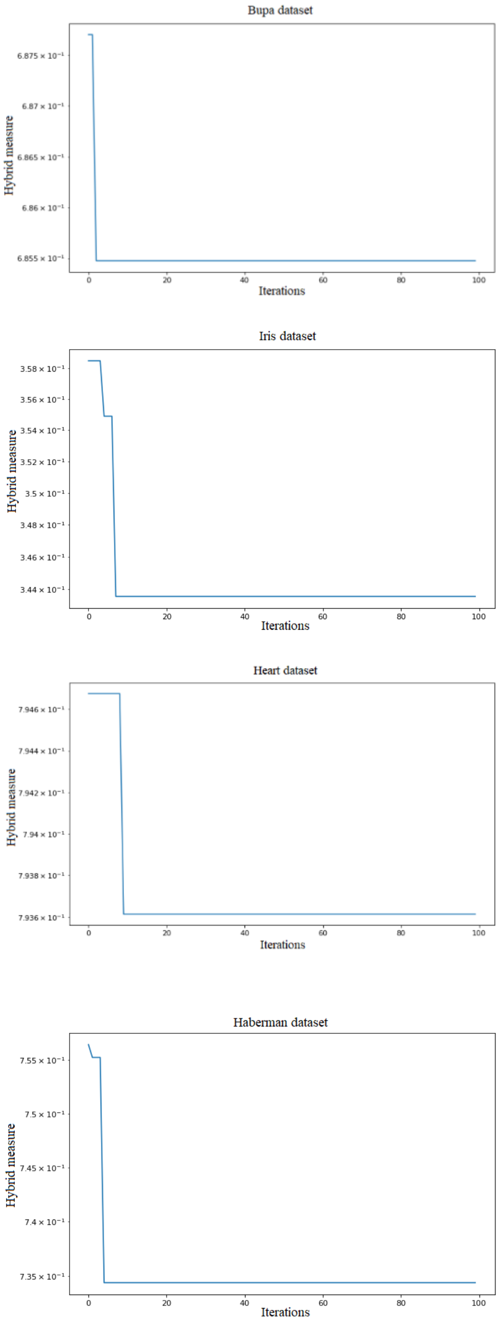


Figure 2: The curve of the VLSMO with the hybrid score on five tested datasets, indirect representation

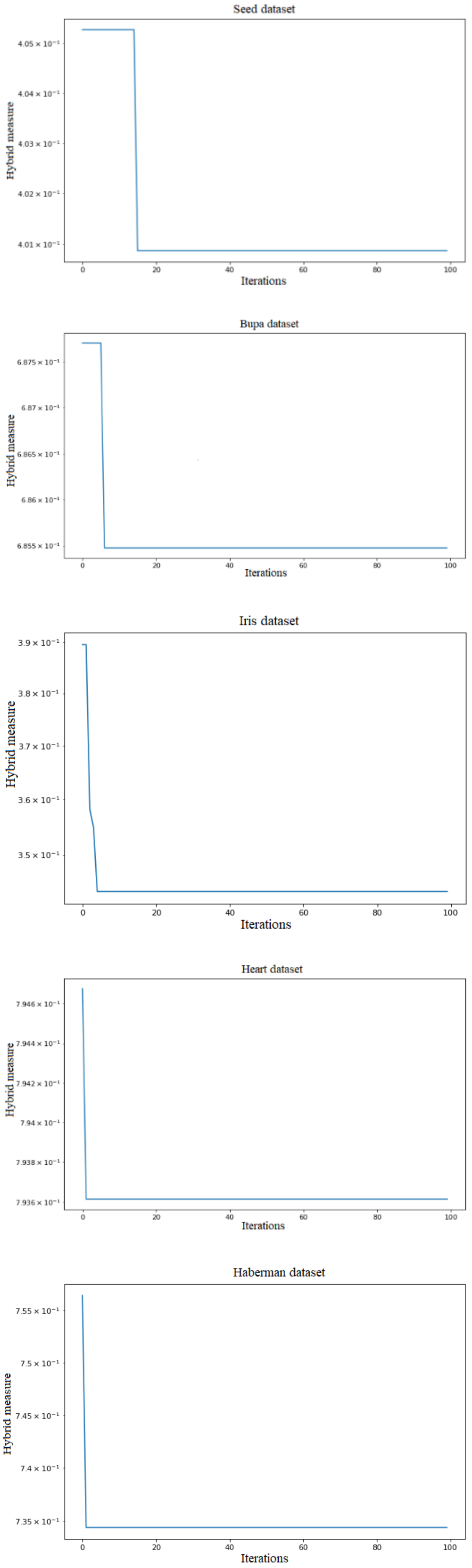


Figure 3: The curve of the VLSMO with the hybrid score on five tested datasets, direct representation

5.3.3 Experiment III

In this experiment, we compared the effectiveness of the VLSMO method against K-Means, PSO, Hybrid, K-Means+GA, and a hybrid sequential clustering algorithm. The results, summarized in Table 16, present the quantization error (average distance between data points and cluster centroids) for the artificial problem I, artificial problem II, Wine, and Iris datasets. The best results are highlighted in bold, indicating superior performance.

Table 16: Comparison of VLSMO with K-Means, PSO, Hybrid, K-Means+GA, and a hybrid sequential clustering (HSC) algorithm regarding quantization error.

Algorithm	**Artificial problem I**	**Artificial problem II**	**Wine**	**Iris**
K-Means	0.984 ±0.032	0.264 ±0.001	1.139 ±0.125	1.139 ±0.125
PSO	0.769 ±0.031	0.252 ±0.001	1.493 ±0.095	0.774 ±0.094
Hybrid	0.768 ±0.048	0.250 ±0.001	1.078 ±0.085	0.633 ±0.143
KMeans+ GA	0.772 ±0.05	0.260 ±0.001	1.384 ±0.099	0.982 ±0.128
HSC	0.764 ±0.031	0.250 ±0.001	**1.072 ±0.084**	0.628 ±0.092
VLSMO	**0.761± 1.9135e-15**	**0.250 ±4.432e-15**	1.074± 1.3688e-16	**0.624± 1.2243-15**

6 Discussion

The results of Experiment I (Tables 3-13) reveal that the proposed hybrid measure, HS, outperforms all other measures. This finding suggests that combining Silhouette and ARI measures into a hybrid measure can address the limitations of each individual measure, resulting in a more robust and comprehensive evaluation of clustering performance. The HS measure considers both the similarity of objects within clusters and the agreement between actual and predicted cluster labels, making it particularly useful for assessing clustering algorithms on complex and heterogeneous datasets. As a result, the HS measure will be utilized in the experiment II.

In Experiment II, the results indicate that employing the VLSMO algorithm leads to superior performance compared to traditional techniques for setting parameters of k-means. The VLSMO algorithm surpasses the basic k-means algorithm to achieve the best value, mean, standard deviation, and speed in identifying the optimal value for the tested clustering datasets. Figures 2-3 demonstrate that the direct technique exhibits faster convergence with VLSMO than the indirect technique.

Moving on to Experiment III, the outcomes consistently demonstrate that the VLSMO algorithm achieves the lowest average quantization error across all tested datasets, except for the Wine dataset, where the Hybrid Sequential clustering algorithm performs better. While other algorithms may yield better results in specific cases, their overall performance lacks consistency. Only the proposed VLSMO algorithm consistently generates the best solutions across the board.

These findings highlight the efficacy of the VLSMO algorithm in various experimental scenarios, supporting its suitability for optimizing clustering outcomes. The consistent superior performance of the VLSMO algorithm in terms of quantization error underscores its potential for effectively addressing complex clustering challenges.

7 Conclusion

In this paper, we proposed a new version of the SMO algorithm called VLSMO, which enhances its flexibility by eliminating the constraint of fixed-length individuals. Unlike the basic SMO algorithm, VLSMO can handle both fixed and variable-length individuals, enabling it to tackle problems that were previously intractable. This feature makes the VLSMO algorithm suitable for solving a variety of problems that the SMO algorithm alone cannot address.

To evaluate the effectiveness of the VLSMO algorithm, we applied it to significant clustering problems involving the identification of optimal hyperparameters for the k-means algorithm, specifically the optimal value of k (number of clusters) and the initial centroids. While existing techniques like the elbow method, gap statistic, and canopy method are used to determine the appropriate cluster size, and methods such as the Forgy method, random partition method, and k-means++ algorithm are employed to find initial centroids, the VLSMO algorithm demonstrates efficient and accurate identification of both optimal values for k and initial centroids, as demonstrated by the test results.

We aim to extend the application of the VLSMO algorithm to a wide range of problems that involve variable-length individuals. Some examples include scheduling time-based sensor networks [21], optimizing road traffic coordination in a multipath scenario [22], and other similar issues.

References

[1] I. Aljarah, H. Faris, and S. Mirjalili, Evolutionary data clustering: Algorithms and applications. Springer, 2021, https://doi.org/10.1007/978-981-33-4191-3.

[2] 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, 2022, https://doi.org/10.11591/ijece.v12i5.pp5435-5443.

[3] C. Yuan and H. Yang, "Research on K-value selection method of K-means clustering algorithm," J, vol. 2, no. 2, pp. 226-235, 2019, https://doi.org/10.3390/j2020016.

[4] S. Saatchi and C. C. Hung, "Hybridization of the ant colony optimization with the k-means algorithm for clustering," in Image Analysis: 14th Scandinavian Conference, SCIA 2005, Joensuu, Finland, June 19-22, 2005. Proceedings 14, 2005: Springer, pp. 511-520, https://doi.org/10.1007/11499145_52.

[5] A. Kumar, D. Kumar, and S. Jarial, "A novel hybrid K-means and artificial bee colony algorithm approach for data clustering," Decision Science Letters, vol. 7, no. 1, pp. 65-76, 2018, https://doi.org/10.5267/j.dsl.2017.4.003.

[6] M. Neshat, S. F. Yazdi, D. Yazdani, and M. Sargolzaei, "A new cooperative algorithm based on PSO and k-means for data clustering," Journal of Computer Science, vol. 8, no. 2, p. 188, 2012, https://doi.org/10.3844/jcssp.2012.188.194.

[7] 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.

[8] Y. Li, Z. Ni, F. Jin, J. Li, and F. Li, "Research on clustering method of improved glowworm algorithm based on good-point set," Mathematical Problems in Engineering, vol. 2018, 2018, https://doi.org/10.1155/2018/8724084.

[9] Z. Bin, G. Zhichun, and H. Qiangqiang, "A Genetic Clustering Method Based on Variable Length String," in 2019 2nd International Conference on Safety Produce Informatization (IICSPI), 2019: IEEE, pp. 460-464, https://doi.org/10.1109/iicspi48186.2019.9095977.

[10] G. Komarasamy and A. Wahi, "An optimized K-means clustering technique using bat algorithm," European Journal of Scientific Research, vol. 84, no. 2, pp. 263-273, 2012.

[11] T. Hassanzadeh and M. R. Meybodi, "A new hybrid approach for data clustering using firefly algorithm and K-means," in The 16th CSI international symposium on artificial intelligence and signal processing (AISP 2012), 2012: IEEE, pp. 007-011, https://doi.org/10.1109/aisp.2012.6313708.

[12] G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," Applied mathematics and computation, vol. 217, no. 7, pp. 3166-3173, 2010, https://doi.org/10.1016/j.amc.2010.08.049.

[13] S. F. Raheem and M. Alabbas, "Fuzzy logic-based self-adaptive artificial bee colony algorithm," in AIP Conference Proceedings, 2023, vol. 2591, no. 1: AIP Publishing, https://doi.org/10.1063/5.0119873.

[14] D. Karaboga and B. Akay, "A modified artificial bee colony (ABC) algorithm for constrained optimization problems," Applied soft computing, vol. 11, no. 3, pp. 3021-3031, 2011, https://doi.org/10.1016/j.asoc.2010.12.001.

[15] M. Alabbas and A. Abdulkareem, "Hybrid artificial bee colony algorithm with multi-using of simulated annealing algorithm and its application in attacking of stream cipher systems," Journal of Theoretical and Applied Information Technology, vol. 97, pp. 23-33, 01/15 2019.

[16] 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.

[17] K. P. Sinaga and M.-S. Yang, "Unsupervised K-means clustering algorithm," IEEE access, vol. 8, pp. 80716-80727, 2020, https://doi.org/10.1109/access.2020.2988796.

[18] 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, 2022, https://doi.org/10.31449/inf.v46i6.4203.

[19] S. F. Raheem and M. Alabbas, "Dynamic Artificial Bee Colony Algorithm with Hybrid Initialization Method," Informatica, vol. 45, no. 6, 2021, https://doi.org/10.31449/inf.v45i6.3652.

[20] C. Blake and C. Merz, "UCI repository of machine learning databases, 1998).(http," archive. ics. uci. edu/ml/index. php.

[21] V.-P. Ha, T.-K. Dao, N.-Y. Pham, and M.-H. Le, "A variable-length chromosome genetic algorithm for time-based sensor network schedule optimization," Sensors, vol. 21, no. 12, p. 3990, 2021, https://doi.org/10.3390/s21123990.

[22] L. Cruz-Piris, I. Marsa-Maestre, and M. A. Lopez-Carmona, "A variable-length chromosome genetic algorithm to solve a road traffic coordination multipath problem," IEEE Access, vol. 7, pp. 111968-111981, 2019, https://doi.org/10.1109/access.2019.2935041.

[23] S. Rana, S. Jasola, and R. Kumar, "A hybrid sequential approach for data clustering using K-Means and particle swarm optimization algorithm," International Journal of Engineering, Science and Technology, vol. 2, no. 6, pp. 167-176, 2010, https://doi.org/10.4314/ijest.v2i6.63708.