

Data Clustering using Two-Stage Eagle Strategy Based on Slime Mould Algorithm

¹Rachid Oucheikh, ²Achraf Touil and ³Mouhsene Fri

¹Department of Physical Geography and Ecosystem Science, Lund University, Lund, Sweden

²Euromed Institute of Technology, Euromed University of Fes, Morocco

³Department of GEGM (Mechanical Engineering and Electrical Engineering), Hassan I University, Settat, Morocco

Article history

Received: 13-06-2022

Revised: 09-09-2022

Accepted: 14-09-2022

Corresponding Authors:

Mouhsene Fri

Euromed Institute of

Technology, Euromed

University of Fes, Morocco

Email: m.fri@ueuromed.org

Abstract: Data clustering is considered an important component of data mining which aims to split a given dataset into disjoint groups having the same similarities. The developed techniques for clustering have some challenges to cluster entities in complex search space and most of them aim to maximize the sum of inter-cluster distances and minimize the sum of intra-cluster distances. This objective function is nonlinear and hard to optimize especially for complex search space. Metaheuristics are becoming a trend for solving this task thanks to their promising results. In this study, the eagle strategy is used to take advantage of the exploration provided by Levy Flight (LF) and the exploitation strength of the Slime Mould Algorithm (SMA) to solve the clustering problem. The SMA algorithm is an efficient technique for solving complex optimization problems which has a high exploitation competence. On the other hand, LF tends to have good exploratory behavior. Our strategy exploits these advantages in a balanced way and through well-designed rounds to ensure the optimality of the clustering solutions. The proposed method is computationally efficient and inexpensive. It also achieves high accuracy in terms of average, worst, best, and the sum of intra-cluster distance. The method is also evaluated according to the speed of convergence and using statistical tests, namely Wilcoxon. The obtained results are compared with seven benchmarked metaheuristics, namely Grey Wolf Optimizer (GWO), Slime Mould Algorithm (SMA), Whale Optimization Algorithm (WOA), Harris Hawks Optimization (HHO), Sine Cosine Algorithm (SCA), Multi-Verse Optimizer (MVO) and Genetic Algorithm (GA) using eighteen datasets of shapes and UCI repositories.

Keywords: Data Clustering, Clustering Evaluation, Metaheuristic, Eagle Strategy, Slime Mould Algorithm, Levy Flight

Introduction

The volume and speed with which data is generated are dramatically increasing making manual processing beyond the scope of human capacities. The need to deal with this data explosion and extract useful information resulted in many advances in data mining and machine learning. In this context, data clustering emerged as an important approach that provides useful insights into the data and learns its latent features, hence the growing interest shown in this technique by the researchers.

Cluster analysis is a descriptive unsupervised learning technique used to group objects with similar intrinsic properties in disjoint clusters and discover the categories forming a given dataset (a set of observations). This process uses the similarity of measured characteristics as a

discriminating metric to build disjoint clusters such that data points belonging to the same cluster are similar to each other and dissimilar to samples existing in other clusters. Thus, it is the process of discovering the inherent object features and organizing similar ones together.

Formally, if we are tackling the dataset $D = \{d_1, d_2, \dots, d_N\}$, consisting of N objects with F features, then clustering can be defined as a task of grouping objects of the same nature in one group called cluster $C_1 \in C = \{C_1, C_2, \dots, C_K\}$. The common approach is that each object has a unique membership, thus the task is to find an assignment $f: D \rightarrow C$. The issues over there narrow down to two main problems, identifying the optimal number of clusters and correctly determining the cluster of every data point. The total number of combinations in greedy assigning N data points to K groups is:

$$S(N, K) = \frac{1}{K!} \sum_{i=0}^K (-1)^i K - i(k) i^N \quad (1)$$

This shows the complexity of clustering for huge datasets and particularly those having complex shapes and structures. Various clustering algorithms are developed and regardless of their type or the technique used, they aim to maximize coherence within each cluster and dissimilarity among different clusters based on certain similarity metrics. Although current algorithms have several advantages, they still suffer from some problems such as sensitivity to initialization values, computation complexity, and big risk of falling into local optimal solution in addition to the high runtime and incompatibility with complex problems and high-dimensional spaces Mostafaie *et al.* (2020). Recently, nature-inspired metaheuristics have shown great potential to overcome these issues and find global solutions.

Metaheuristic algorithms are optimization tools that are increasingly used to solve complex problems in data mining thanks to some advantages Nayyar and Nguyen (2018); Srinivasu *et al.* (2021); Nayyar *et al.* (2018). They are easy to implement, they do not require information on the objective function gradient and they are generally not vulnerable to local minima. Metaheuristics are used particularly for solving nonconvex clustering problems. The idea is to represent this problem in the optimization domain and formulate the clustering as an optimization approach. The approach aims to find a value x_n within the appropriate n -dimensional search space S that minimizes or maximizes the cost function, i.e., solves an optimization problem.

In this study, a new clustering approach using the eagle strategy over Levy Flight (LF) and Slime Mould Algorithm (SMA) is proposed. This technique tries to find the global optimum and avoid local optima by seeking the best cluster centroids and minimizing some defined clustering metrics. The main contribution of this study is the design of a novel clustering technique using the Eagle Strategy over LF and SMA. To evaluate the performance of the introduced method, we have used eighteen standard benchmark datasets and the results obtained using the approach are compared with seven of the most well-known recently developed algorithms. Moreover, to prove the efficiency of the proposed technique, we have used statistical tests.

Background

Clustering

In this section, we provide the theoretical background and a brief review of previous work done on clustering using metaheuristics. First, some notations and terminology used throughout this study are listed in the following:

- The object is a single data item represented by a vector of measurements $x = \{x_1, x_2, \dots, x_D\}$, where $x_i \in R$ is a feature and D is the object dimension
- A dataset is a set of objects, denoted as
- $X = \{x_1, x_2, \dots, x_N\} \in R_D$, where N is its cardinal representing the total number of objects in the dataset. Thus, a dataset can be represented in the form of a matrix of size $N \times D$: $X = X_n^d$, where $1 \leq n \leq N$ and $0 \leq d \leq D$
- The goal of clustering is to split a dataset X into K of mutually disjoint clusters which is denoted $C = \{c_k | k = 1, \dots, K\}$. Each cluster c_k contains $n_k = |c_k|$ objects
- The centroid of the cluster (or prototype) c_k is expressed as $c_k^- = \frac{1}{n_k} \sum_{x_i \in c_k} x_i$. Similarly, the centroid of data set X is $X^- = \frac{1}{N} \sum_{x_i} X^{x_i}$

Categories of Clustering Algorithms

Clustering techniques can be categorized into two different big categories: Hard clustering which supposes a binary relation between objects and clusters, i.e., each pattern belongs to exactly one cluster, and fuzzy clustering which assumes that an object can belong to one or more clusters and therefore assigns multiple degrees of membership to each object. The first category is more applied, is more interesting in practice, and is the subject of our paper.

More precisely, depending on the used technique, the clustering approaches can be categorized into diverse categories: Partitional, hierarchical, model-based, density-based, and grid-based Shi and Pun-Cheng (2019).

Partitional clustering: aims to decompose the dataset into pre-specified number k disjoint groups starting with initial random k partitions and then refining them using optimization metrics associated with the properties of the search space. The methods belonging to this category optimize the chosen validity indices by iteratively reallocating cluster members. The most well-known techniques of this type are k -means, k -medoid, and CLARA.

Regarding the partitional clustering algorithms inspired by nature, Saemi *et al.* (2016), the authors review some of them and compare their performance on some criteria such as time complexity, stability, and clustering accuracy on real and synthetic data sets.

Hierarchical clustering: Seeks to construct a hierarchy of clusters named a dendrogram tree. The procedure is iterative and produces successive clustering levels according to a similarity metric and follows one of two strategies: Agglomerative or divisive. Agglomerative hierarchical clustering is a "bottom-up" approach that begins by considering each data point as a separate cluster and iteratively identifies and merges the two most similar clusters until all the clusters are merged. Whereas divisive clustering is a "top-down" approach that starts by

considering all the observations as one cluster and then successively splits the obtained clusters. The most well-known algorithms under this category are BIRCH, CURE, ROCK single-link, and complete-link algorithms.

Model-based methods: Try to find the probabilistic models which fit the data. Starting with the random probabilistic model, the task is to fit this distribution to the given data and optimize the distance between both distributions. The model can be a combination of probability distributions. or its nonparametric formulation Zhang and Di (2020).

Density-based algorithms: Built on the idea that clusters in the dataset's feature space are contiguous areas of high density separated by regions with low densities, called noise or outliers. This clustering approach is particularly suited to cases where clusters have nonlinear shapes and cannot be well described. An example of an algorithm belonging to this category is Dbscan Wang *et al.* (2019).

Grid-based clustering: It partitions the entire data space into separate cells with grids and through data quantization, it merges the cells to construct clusters. A typical example of this method is the Wave Cluster approach Sheikholeslami *et al.* (2000).

Clustering using Metaheuristics

Recently, several researchers develop new metaheuristic optimization algorithms to solve the task of clustering. Metaheuristics are capable to converge to a global optimum while classical clustering algorithms can only ensure finding a local solution. Maulik and Bandyopadhyay (2000) used Genetic Algorithms (GAs) to search for the cluster centers that minimize the sum of the inter-distances between the points inside the cluster and their corresponding centers. Maulik and Saha (2010) proposed a clustering approach based on an evolutionary algorithm, namely differential evolution, and applied it to image classification. Borah and Ghose (2009) proposed a new method based on the Automatic Initialization of Means (AIM) which improves the quality of the clustering results by automating the initial means selection. Swarm optimization is one of the most used methods for performing the clustering approach. Tan *et al.* (2011) evaluated ant colonies for clustering real-world data while the Bees Algorithm is used by the authors (Yan *et al.*, 2012; Karaboga and Ozturk, 2011) to escape the local optima during the process of finding the best cluster centers concerning a defined metric.

An enhanced grey wolf optimizer imitating the behavior of grey wolves adapted with binomial crossover is proposed for clustering (Tripathi *et al.*, 2018). Various variants of this technique are designed to increase its performance and convergence speed such as Bees Algorithm with a memory scheme in which a memory component is used to avoid visiting sites that have the

same fitness or worse and which are close to previously visited sites (Nemmich *et al.*, 2018). Janani and Vijayarani (2019) used particle swarm optimization for text document clustering. Harris hawk's optimization algorithm which is inspired by the cooperative behavior and chasing style of Harris' hawks in nature is recently used for data clustering (Singh, 2020). The slime mold technique is used recently for clustering, but only for the improvement of prediction accuracy achieved by k-means (Chen and Liu, 2020). The objective of slime mold was to renew the population and optimum parameters which are used by the support vector machine and k-means.

Some papers provide a general review of the clustering based on different metaheuristics (Bagirov *et al.*, 2020). Nanda and Panda (2014) reviewed the clustering approaches based on nature-inspired metaheuristic algorithms. Abualigah *et al.* (2020) presented a detailed survey of nature-inspired metaheuristics for text document clustering. Particularly, Esmine *et al.* (2015) provided a thorough survey of high-dimensional clustering techniques which are based on particle swarm optimization and its variants. Metaheuristics techniques are easy to implement and can be applied directly to the clustering problem or be applied to the output of another clustering method to improve the quality of the obtained solutions. They are designed to optimize an objective function, called fitness function, that guides progressively the search. However, they are less accurate and may require significant computational power, particularly for large datasets.

In our paper, we are concerned with partitional clustering which is considered a global optimization problem. It is worth mentioning that traditional clustering algorithms such as k-means can only guarantee convergence to a local solution. The obtained local solution may not yield a meaningful and real structure of the studied data. In addition, clustering algorithms based on branch and bound methods and dynamic programming are applicable only in datasets with small sizes. Therefore, the use of smart metaheuristics is necessary to overcome these limitations.

Using eighteen benchmarked datasets, extensive experimental results are performed in Ezugwu (2020) on main nature-inspired metaheuristics that have been developed for automatic clustering. The comparisons and statistical significance showed that all the considered clustering algorithms can effectively provide accurate clustering partitions and determine the optimal number of clusters. But the firefly algorithm and its hybrid clustering techniques provide high-quality solutions and outperform all of them in both high and low-dimensional data spaces. Despite the merits and strengths of the mentioned techniques, they still face some challenges. The most apparent of which is the problem of the local optima trap.

Moreover, a well-known theorem in the domain of heuristics named No Free Lunch Adam *et al.* (2019) states that there is no single optimization algorithm capable of solving all accurately and with high precision all problems. As an implication, it is always possible that any algorithm which performs well on a data set may do worse on another. For all these reasons, it is interesting to develop new techniques and investigate their efficiencies for data clustering. It is in this context that our contribution takes place.

Materials and Methods

Validity Indices

To compare and measure the performance of the algorithms on a dataset, several metrics can be used to measure the similarity between the obtained clusters. The notion of distance is very important for clustering tasks and helps to measure the similarity among data objects and clusters. To measure the distance, a metric (or quasi-metric) is used to quantify the proximity between objects. These indices, referred to as Cluster Validity Indices (CVI), evaluate particularly the quality of a specific clustering structure obtained by a clustering algorithm. They should be easy to compute and meaningful in the way of defining the relationships between clusters (inter-groups) and data points (intra-group). There are two classes of clustering metrics: Internal and external. Evaluation with internal metrics is based only on the data used for clustering, whereas evaluation with external metrics is made against unseen data that was not used for clustering, generally with known cluster labels.

The clustering problem can be formulated as an optimization problem that could be solved by single-objective and multiobjective metaheuristics. Most metaheuristic approaches use internal validity indices as function objectives to optimize. The core idea of these metrics is to assign the best score to an algorithm that returns a partition with high similarity within a cluster and low similarity between clusters without any ground-truth data (external information). A detailed study with experiments on the properties of clustering was carried out by Arbelaitz *et al.* (2013). But recently, Hämäläinen *et al.* (2017) compared the behavior of different CVIs and empirically evaluate some of the internal clustering validation indices with many datasets. As a result of the comparison, the author state that the selection of CVI depends on the shape of the dataset and that there is no perfect CVI. The clustering function objective can be expressed in terms of one or more of the internal validity indices such as centroid distance, intra, and inter-clustering distance, Silhouette, variance ratio criterion, distortion distance, Dunn's index, medoid distance, CS measure, Davies-Bouldin index.

External indexes require ground-truth data. Some of them, such as the rand index, Jaccard coefficient, and Fowlkes and Mallows index assess and count the co-location of pairs of data points into the same (or different) cluster in the solution against the ground truth. Other external metrics are cluster-level and estimate the similarity of two cluster solutions.

In this study, the clustering task is formulated as an optimization problem where the goal is to minimize the fitness function expressed as the Within-Cluster Sum of Squares (WCSS):

$$F(X, C) = \sum_{C=1}^C \sum_{x_i \in C} \|x_i - \bar{c}_i\|^2 \quad (2)$$

where $\|\cdot\|^2$ denotes the L²-norm between the two vectors x_i indicating the datapoint and \bar{c}_i representing the centroid of the cluster i .

Eagle Strategy

Eagle Strategy (ES) is an optimization strategy inspired by the hunting behavior of golden eagles or Aquila Chrysaetos. An eagle starts foraging by flying freely and in a random way and searching globally the prey. Once the prey is seen, the eagle changes its search manner and adopts intensive hunting to efficiently seize the prey. This behavior can be summarized in two successive important processes: hunting strategy and chase tactics.

Inspired by this behavior, Yang and Deb (2010) proposed the Eagle Strategy (ES) that combines in balanced way different algorithms to efficiently carry out both processes. Accordingly, ES has two stages. The first stage performs a crude global search reflecting the fact that the eagle starts searching globally for prey. The second phase performs an intensive local search around the promising solution or the set of promising solutions and reflects the chase of the target made by the eagle.

The use of suitable algorithms for each phase speeds up the convergence to the solution and enhances the quality of the obtained solution. ES is similar to other algorithms (such as Random-restart hill climbing) which use the random restart mechanism so that to escape from local minima of the fitness function and consequently improve their performance. They try first to generate a good starting point from which the search starts until achieving a solution. Then, the solution is evaluated and if it looks not accurate, another point should be selected and considered as a starting point for a new iteration of the search. However, there are some essential characteristics of ES. First, it is not a method, but rather a strategy. Second, various algorithms can be used within their different stages. Third, its two stages can be toggled on and off depending on the quality of the found solutions.

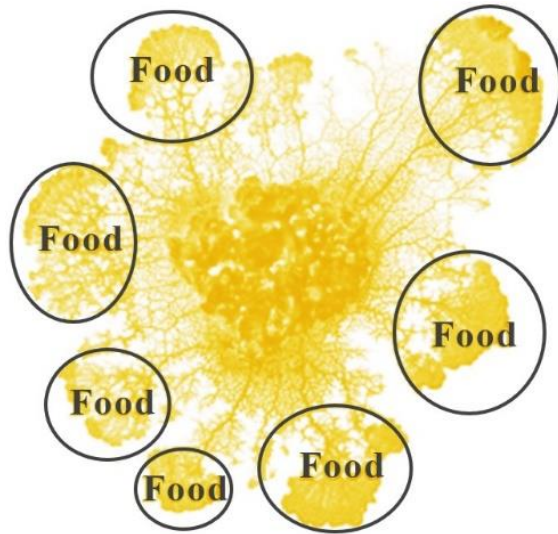


Fig. 1: Slime mold foraging

Levy Flight

Levy Flight (LF) is a heavy-tailed probability distribution proposed by Paul Levy. It is a non-Gaussian random process inspired by random walk Chechkin *et al.* (2008) but with greater steps. The lengths of individual jumps are distributed with the Probability Density Function (PDF) $\lambda(x)$ decaying at large x as $\lambda(x) = |x|^{-1-\alpha}$.

The occurrence of extremely long jumps is due to the divergence of LF variance $\langle x^2(t) \rangle \rightarrow \infty$ and the trajectories include shorter jumps interspersed by long excursions:

$$x_j^{new} = x_j + a \otimes \text{levy}(\beta) \quad j = 1, 2, \dots, NG \quad (3)$$

where: α is the random step size parameter, β is the Levy flight distribution parameter, and \otimes is entry-wise multiplication where α is the step size; with $0 < \alpha < 2$, L is a value from the Levy distribution, $j = 1, 2, \dots, n$ is the number of nests considered. In our approach, the levy flight technique helps to increase the exploration and quickly converge toward the solution.

Slime Mould Algorithm

Slime mold is a new metaheuristic inspired by the *Physarum polycephalum*. Slime mold consists of several types of unrelated eukaryotic organisms that live in cold and humid places. These organisms are single cells, but they can unite and join so to constitute multicellular reproductive beings. There are several techniques inspired by the pattern and behaviors of slime mold. Li *et al.* (2011) proposed routing protocols that can be used in wireless sensor networks, and which are inspired by slime mold foraging.

The proposed routing protocols enhance the tradeoff between efficiency and robustness. Yang *et al.* (2019) proposed a new transportation network design inspired by slime mold foraging. The proposed model aims to connect 31 nodes which represent 31 cities in China with higher performance in cost and stability. Zhang *et al.* (2016) adopt the mathematical pattern of slime mold foraging to design a supply chain network of oligopolistic firms. Compared to Nagurney (2010), the solution shows an efficient practical result. Thus, slime mold is mostly used to model the networks and to resolve the problems that can be formulated as graphical objects. However, in this study, we will use the search behavior of slime mold and its intelligent food-see mechanism for optimization goals. To seek food, slime mold first forms a network venous through its organic matter and takes advantage of the humid condition. This organic matter spread steadily to contact food. Due to the nature of the network, as seen in Fig. 1, the slime mold can find different foods in different places at the same time. When slime mold comes in contact with a food source, the bio-oscillator sends a propagating wave from the outer legs inward. The variation of this wave depends on the quality of the found food. When the slime mold receives a strong wave, it reinforces the cytoplasmic flow through the vein Nakagaki *et al.* (2000). Therefore, the diameter of the path in contact with a high-quality food will be larger, the slime mold tends to breed in this area and extensively forage for food in nearby regions Kareiva and Odell (1987). Otherwise, if the slime mold contacts a food of lower quality, the slime mold must rapidly decide to contract the path and change the current position.

This foraging behavior can be formally described in an algorithm with three phases:

Food Searching

Slime mold approaches the food by following the odor in the air and this can be represented as a mechanism of exploration in the algorithm which can be expressed by Eq. 4, where:

- vb = A parameter factor lying in the interval $[-a, a]$, such that a is calculated using Eq. 5
- t = The current iteration
- $\rightarrow -VC$ = A decreasing function from 1 to 0
- $X(t)$ = The actual position of the slime mold
- $X(t+1)$ = The next position of the slime mold
- $X_A(t)$ and $X_B(t)$ = A random future position of slime mold
- P = A probability value expressed by Eq. 6
- W^* = The weight affected by the slime mold for each path

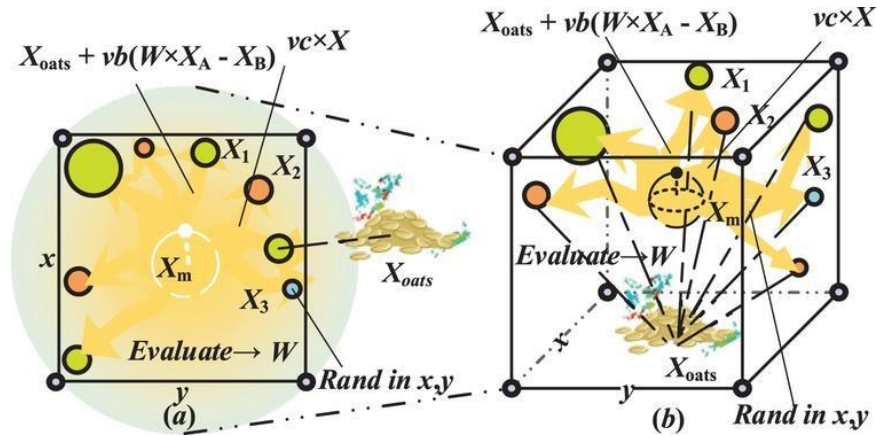


Fig. 2: Possible new positions in 2 and 3-dimension Li *et al.* (2020)

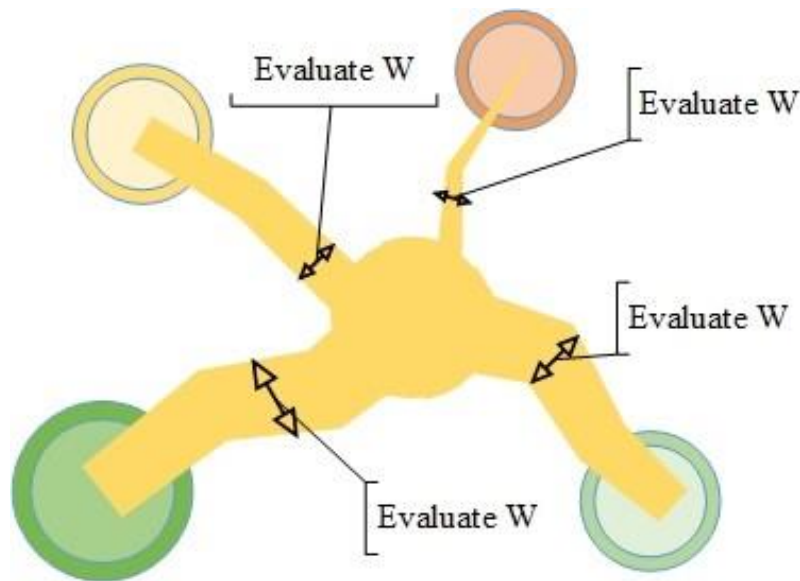


Fig. 3: Assessment of fitness

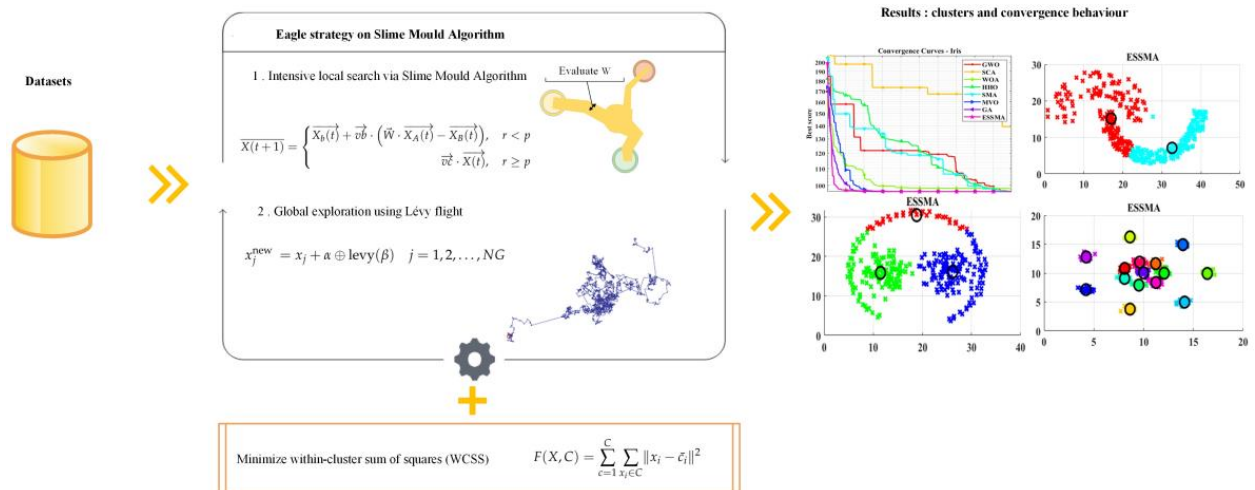


Fig. 4: Flowchart of the proposed approach

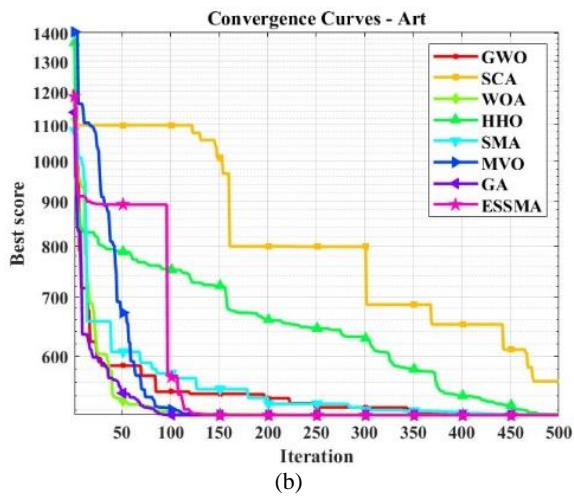
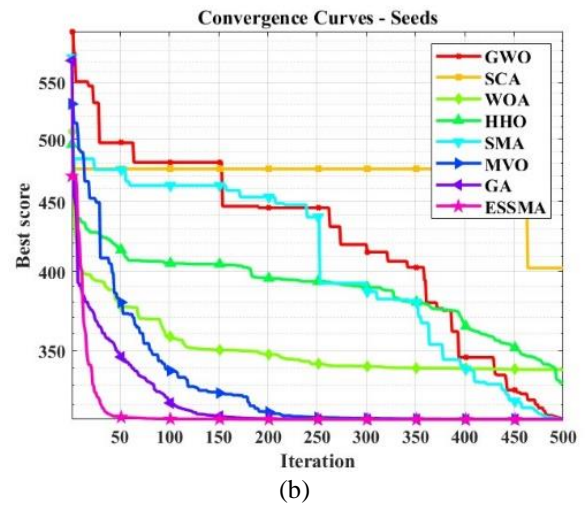
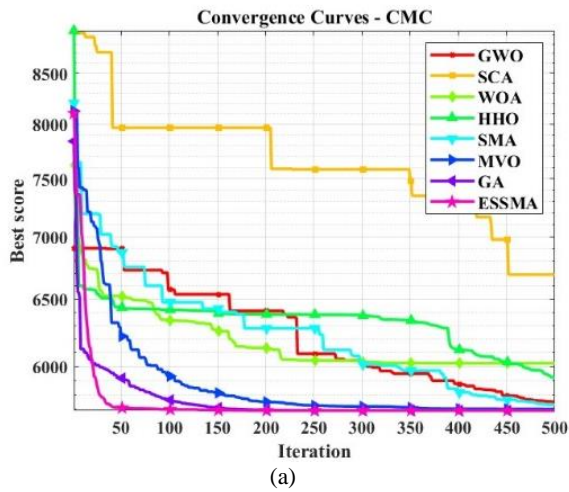


Fig. 6b: Convergence graph of the metaheuristics for a Glass dataset, b Seeds dataset

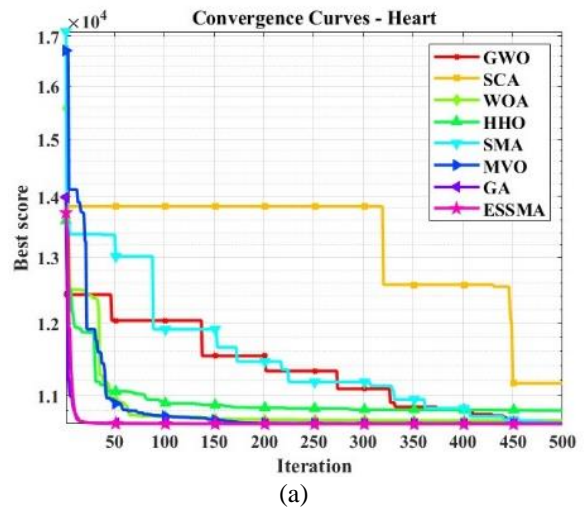


Fig. 5: Convergence graph of the metaheuristics for a CMC dataset, b Art dataset

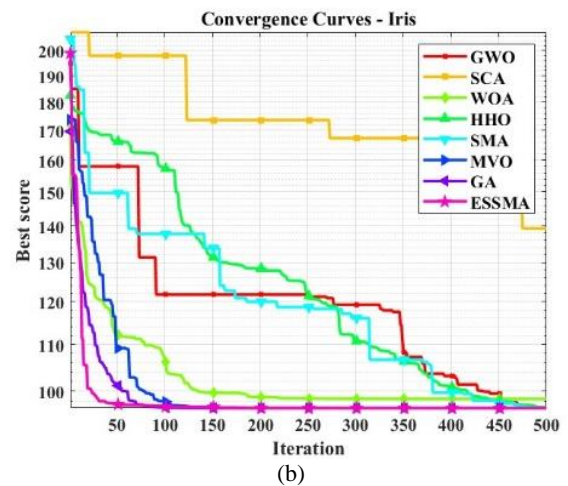
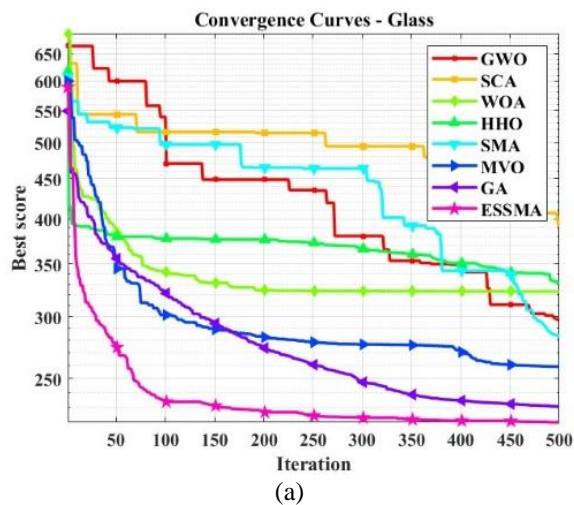


Fig. 6a: Convergence graph of the metaheuristics for a Glass dataset, b Seeds dataset

Fig. 7: Convergence graph of the metaheuristics for a Heart dataset, b Iris dataset

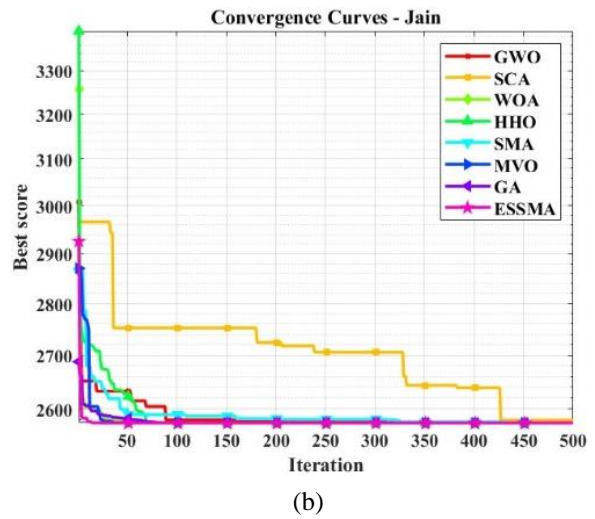
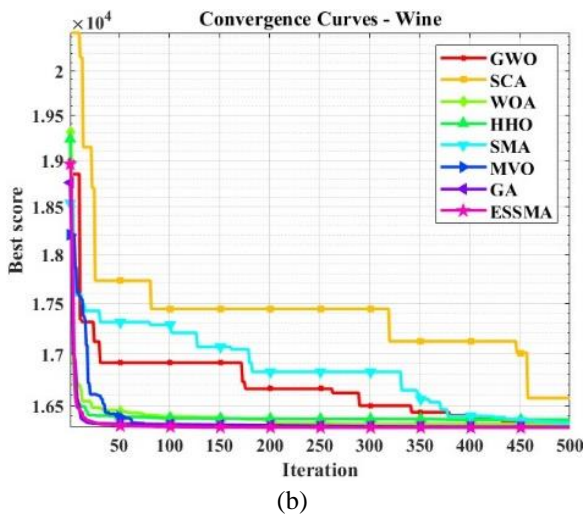
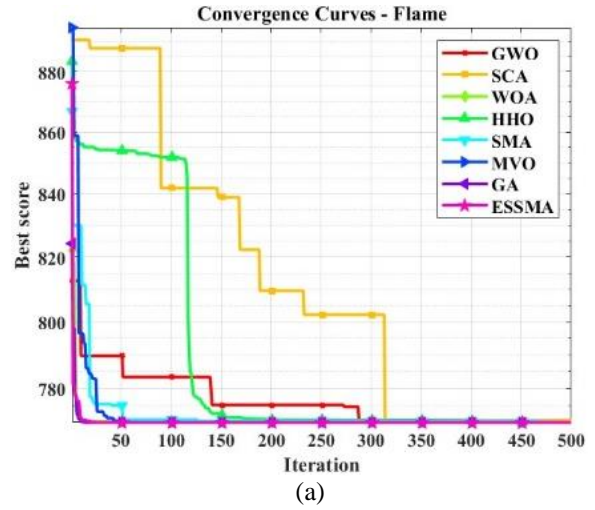
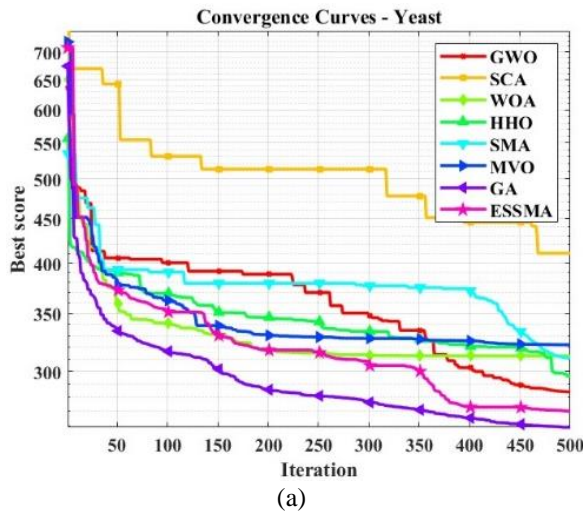


Fig. 8: Convergence graph of the metaheuristics for a Yeast dataset, b Wine dataset

Fig. 10: Convergence of the considered algorithms for a Flame dataset, b Jain dataset

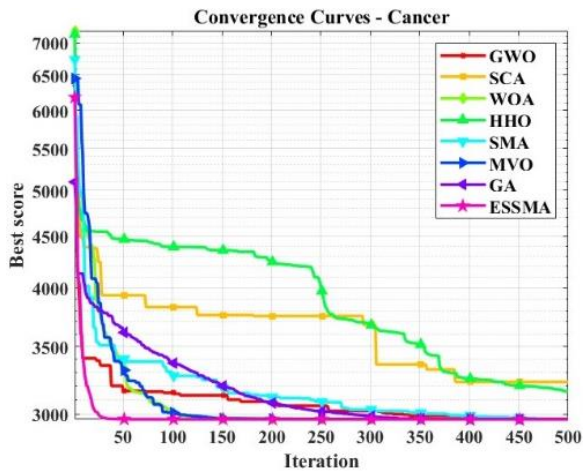


Fig. 9: Convergence graph of the metaheuristics for the Cancer dataset

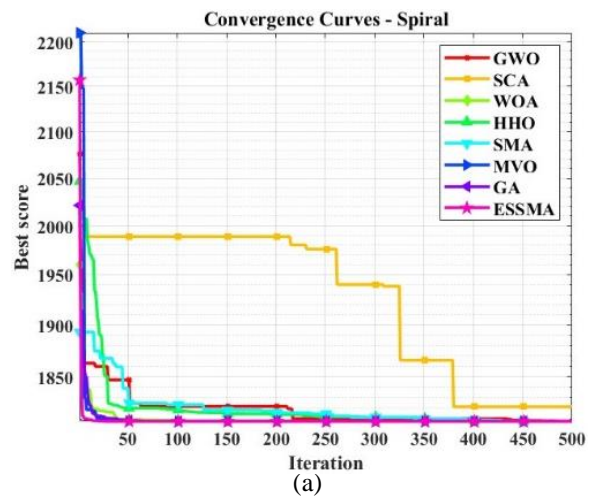
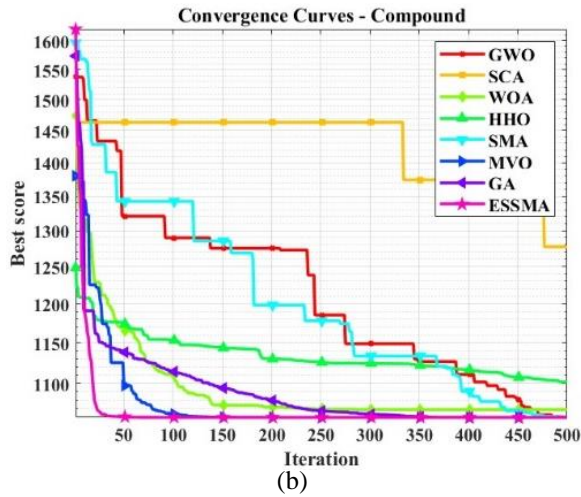
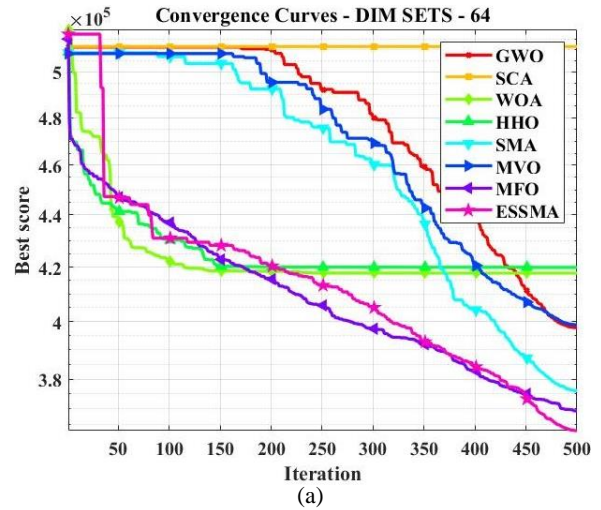


Fig. 11a: Convergence of the considered algorithms for a Spiral dataset, b Compound dataset

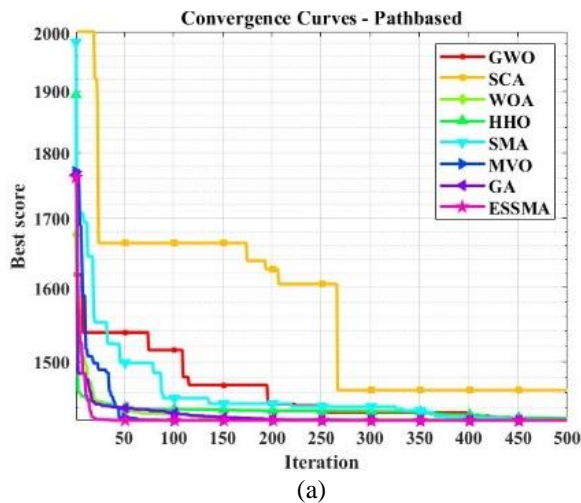


(b)

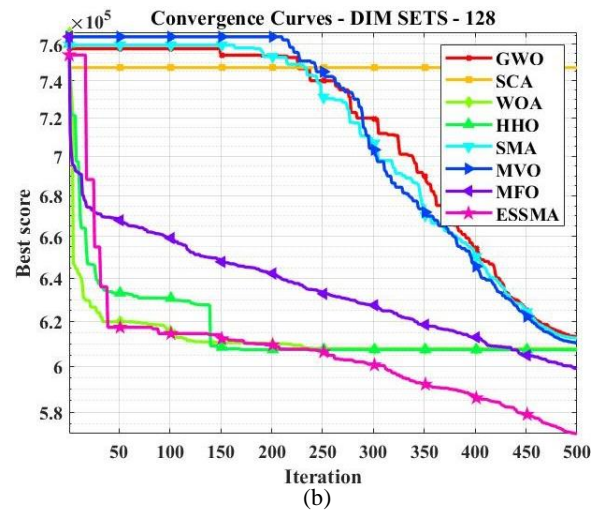
Fig. 11b: Convergence of the considered algorithms for a Spiral dataset, b Compound dataset



(a)

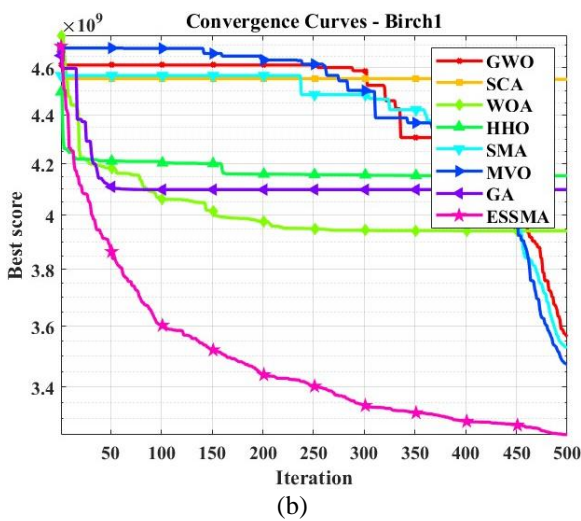


(a)



(b)

Fig. 13: Convergence of the considered algorithms for a DIM64 dataset, b DIM128 dataset



(b)

Fig. 12: Convergence of the considered algorithms for a Path-based dataset, b Birch 1 dataset

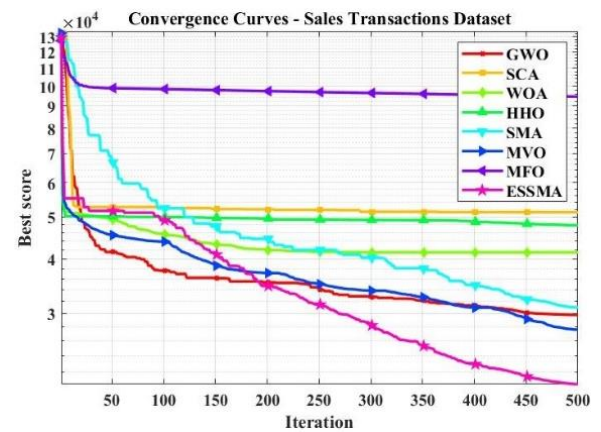


Fig. 14: Convergence of the considered algorithms for the Sales Transactions dataset

$$\overline{X}(t+1) = \begin{cases} \overline{X}_b(t) + vb \cdot (\overline{W} \cdot \overline{X}_A(t) - \overline{X}_B(t)), & r < p \\ vc \cdot \overline{X}(t), & r \leq p \end{cases} \quad (4)$$

Equation 4 describes the processing search inspired by foraging food of slime mold. To explore a new region, the slime mold looks for new promising positions in all directions such as visualized in Fig. 2. The slime mold searches the new positions under conditions of $r < p$ and $r >= p$. If $r < p$, Eq. 5 favor more the exploitation process; the new position is updated according to the best position of $X_B(t)$. Otherwise, the Eq. 6 promotes the exploitation process; the new position is updated according to the tuning value of vc .

$$a = \arctan h \left(- \left(\frac{t}{\max t} \right) + 1 \right) \quad t \in (1, \dots, Max_t) \quad (5)$$

$$p = \tanh | S(i) - DF | \quad i \in 1, 2, \dots, n \quad (6)$$

where, $S(i)$ is the fitness of \vec{X} , DF is the best fitness of \vec{X} :

$$W(\text{SmellIndex}(i)) = \begin{cases} 1 + r \cdot \log \left(\frac{bF - S(i)}{bF - wF} + 1 \right), & \text{condition} \\ 1 - r \cdot \log \left(\frac{bF - S(i)}{bF - wF} + 1 \right), & \text{otherwise} \end{cases} \quad (7)$$

$$\text{Smellindex} = \text{sort}(S) \quad (8)$$

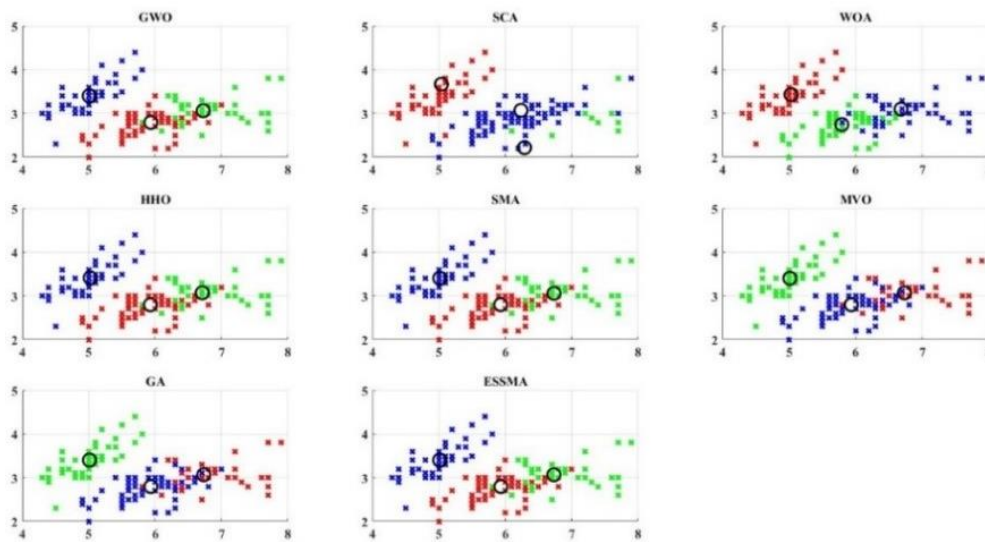


Fig. 15: Clustering results for the Iris dataset

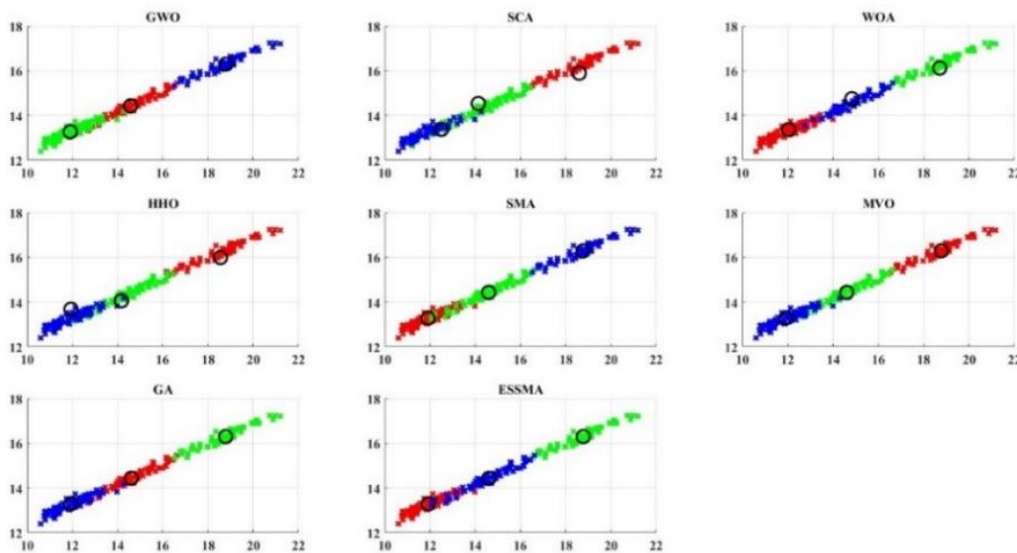


Fig. 16: Clustering results for Seeds dataset

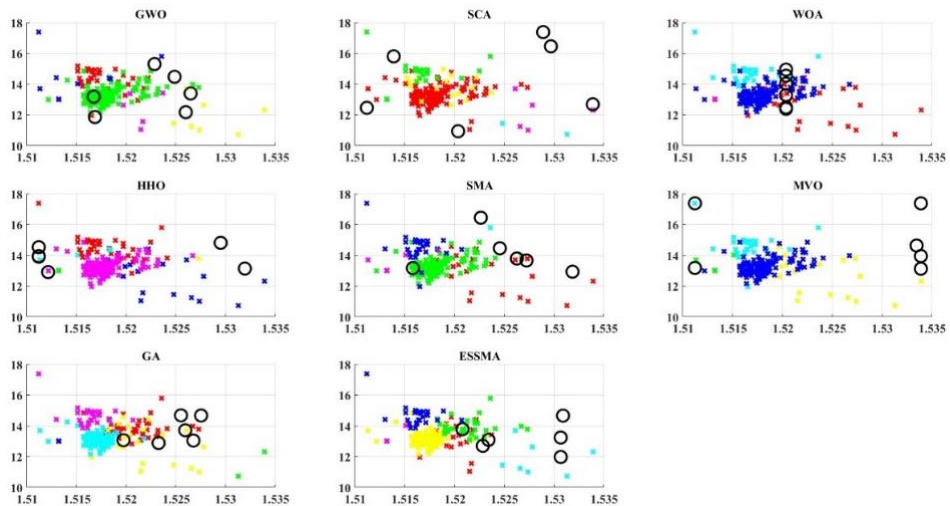


Fig. 17: Clustering results for Glass dataset

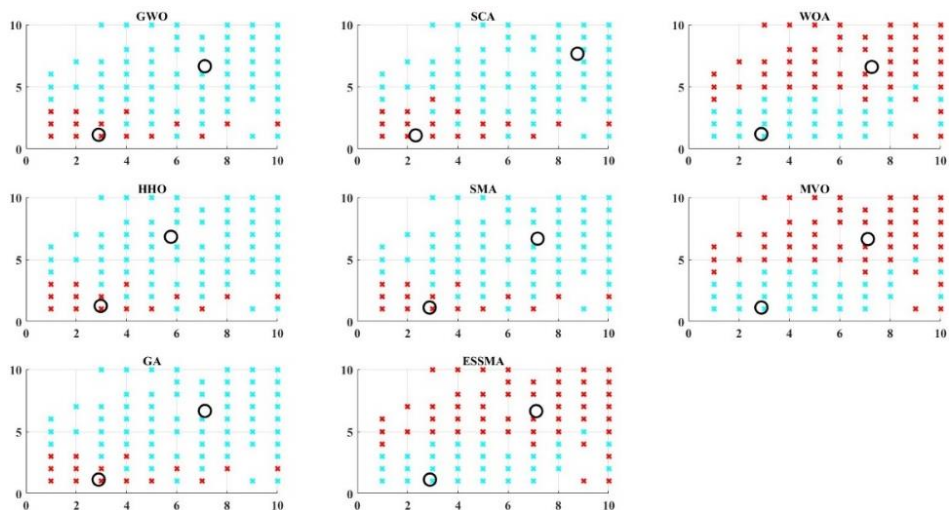


Fig. 18: Clustering results for Cancer dataset

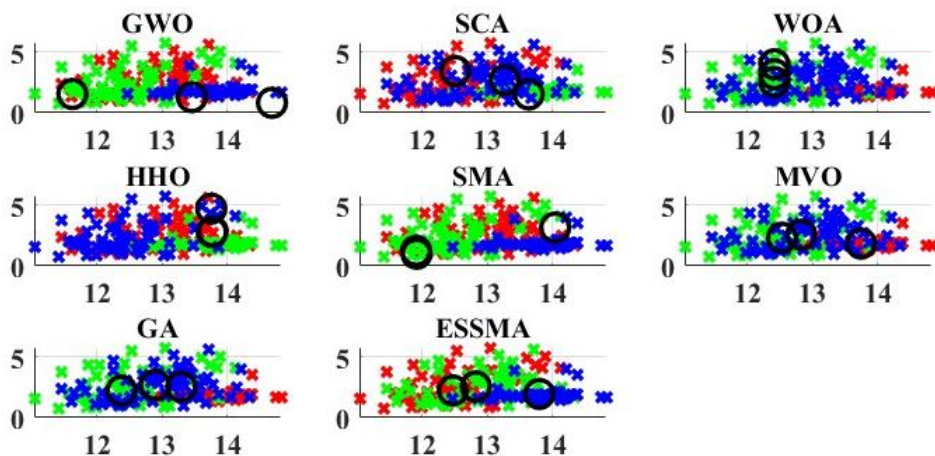


Fig. 19: Clustering results for Wine dataset

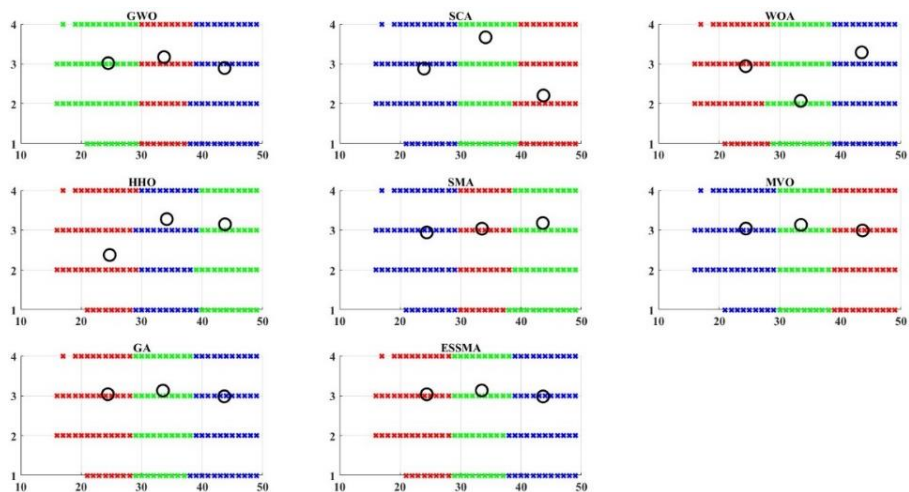


Fig. 20: Clustering results for the CMC dataset

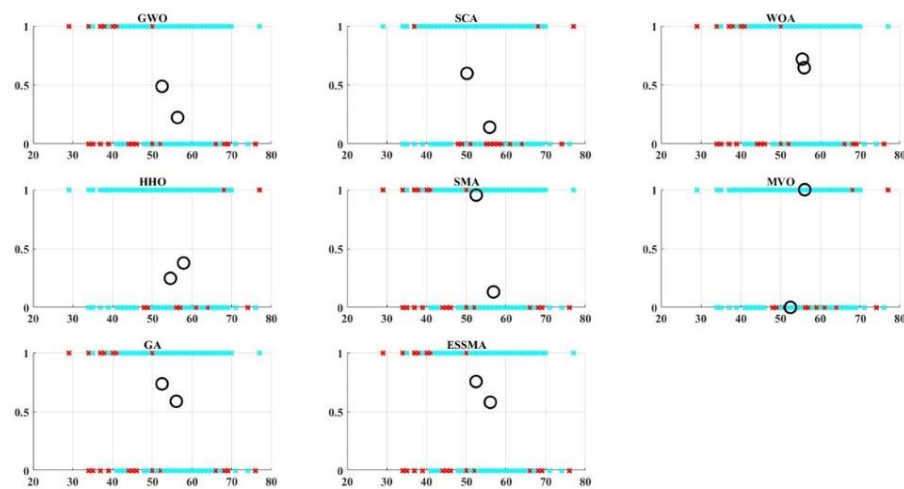


Fig. 21: Clustering results for Heart dataset

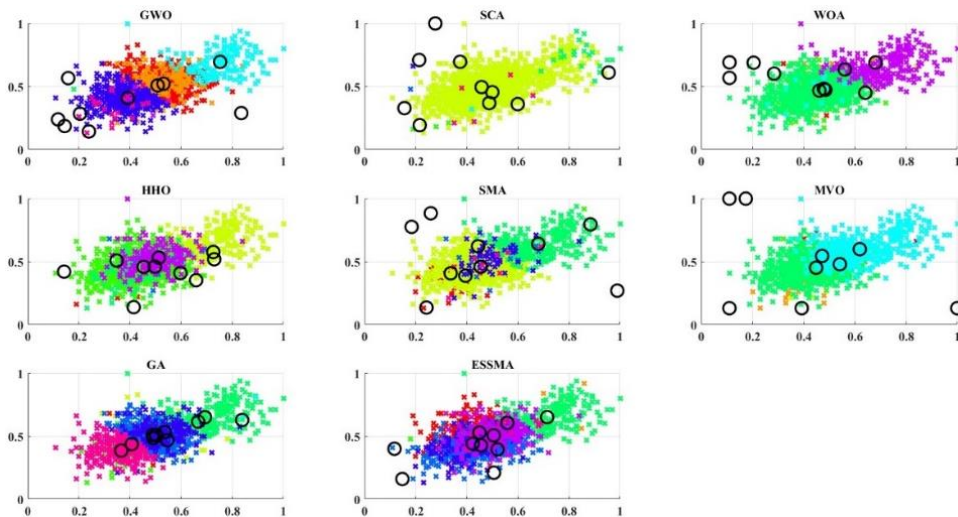


Fig. 22: Clustering results for Yeast dataset

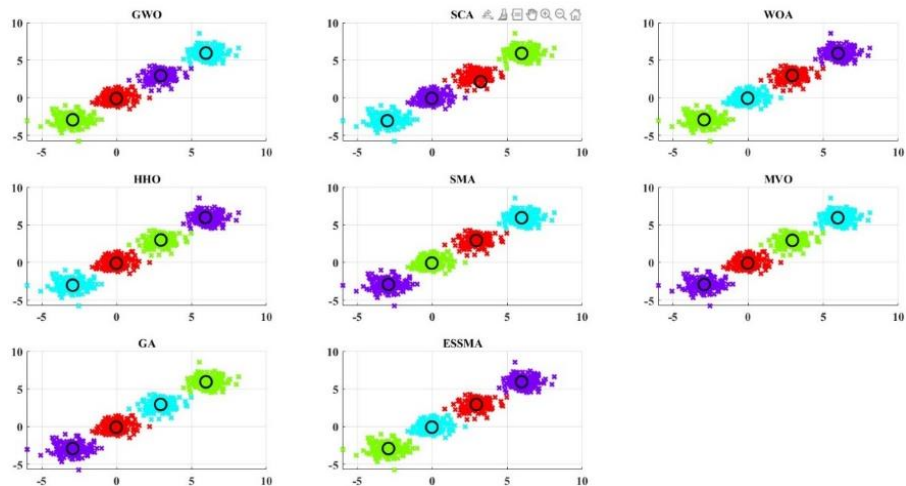


Fig. 23: Clustering results for Art dataset

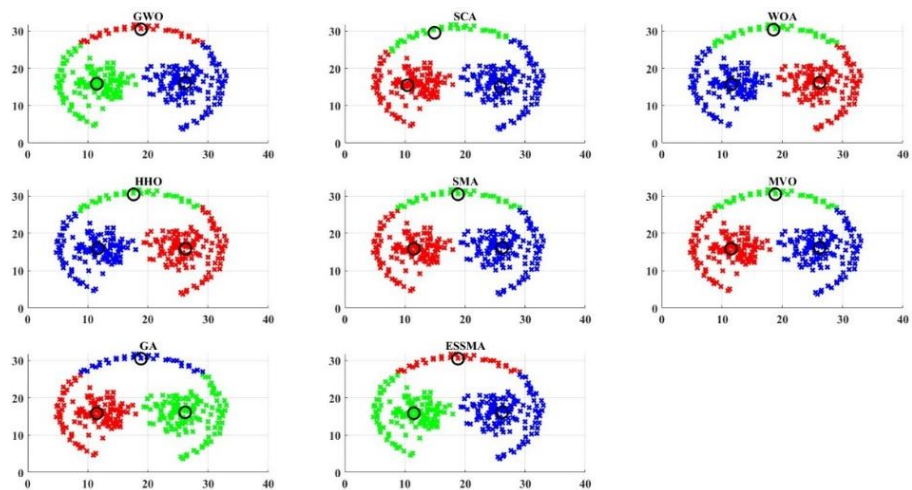


Fig. 24: Clustering results for Path-based dataset

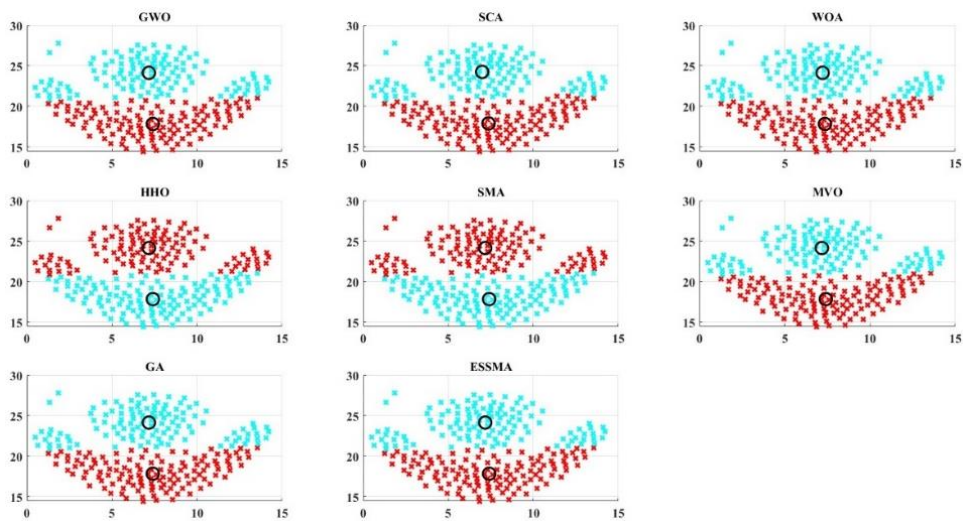


Fig. 25: Clustering results for Flame dataset

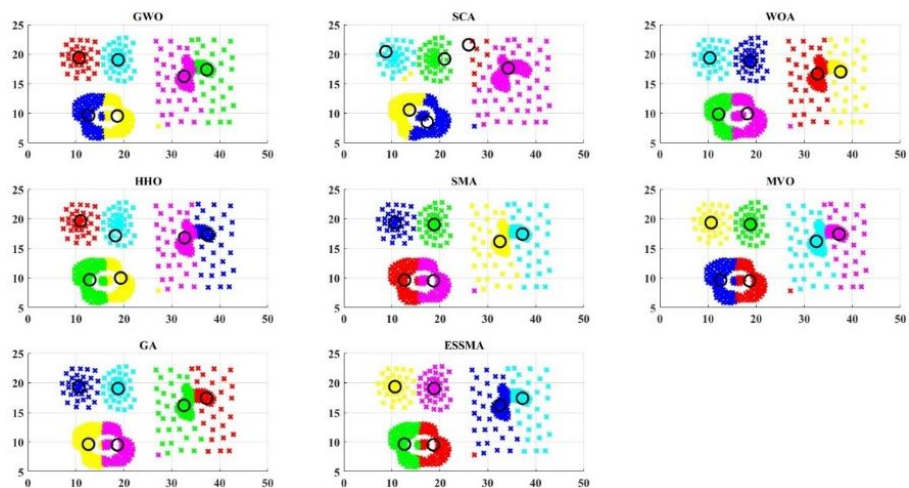


Fig. 26: Clustering results for Compound dataset

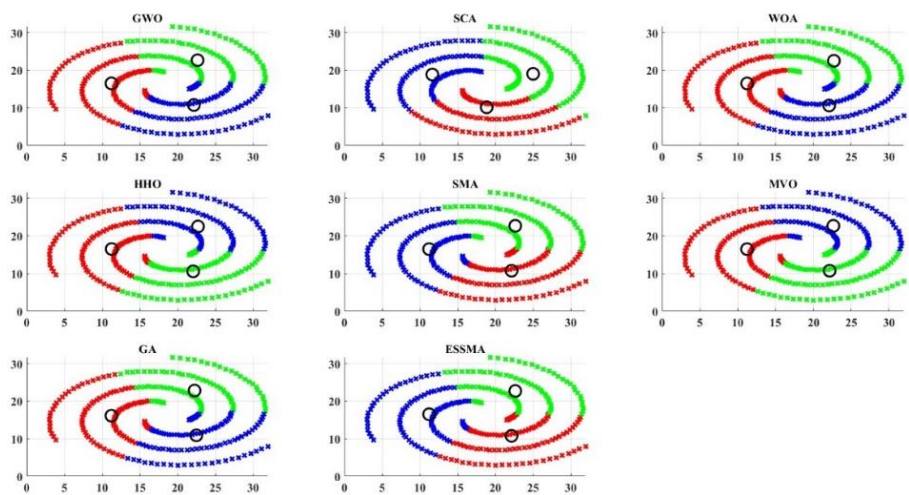


Fig. 27: Clustering results for Spiral dataset

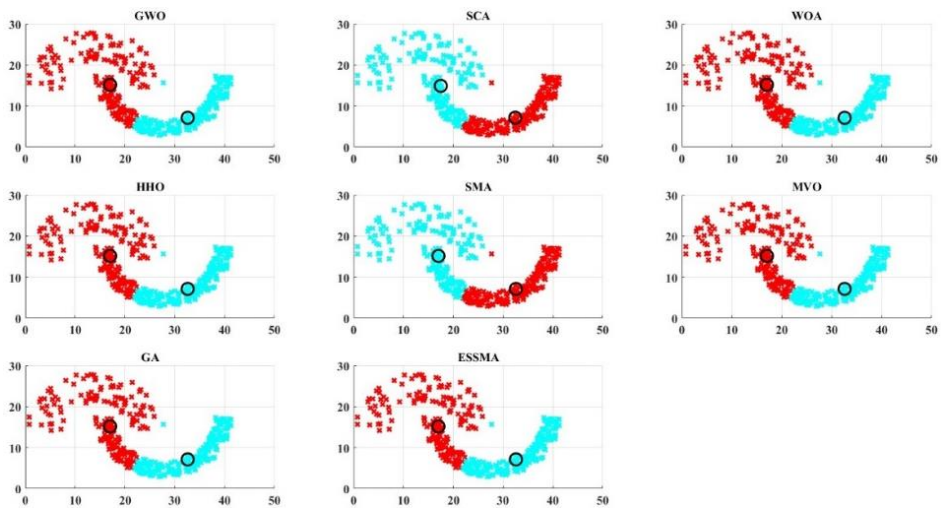


Fig. 28: Clustering results for the Jain dataset

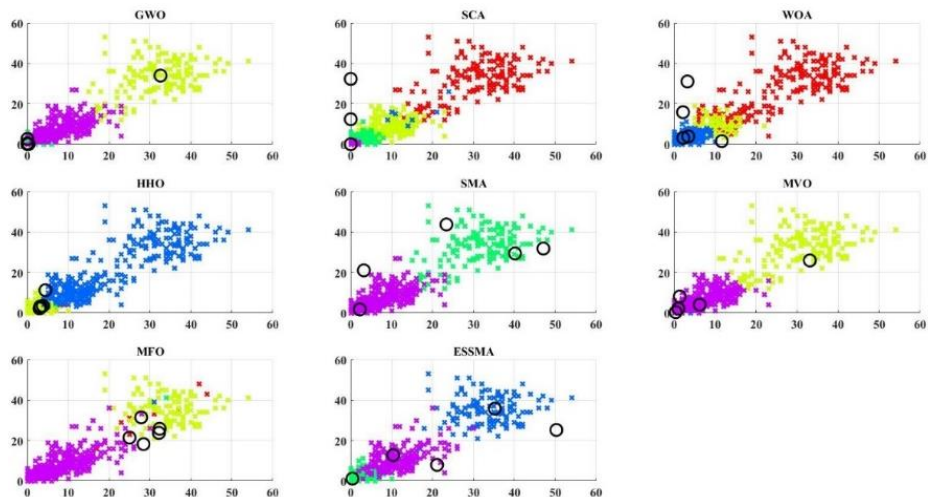


Fig. 29: Clustering results for the Sales Transactions dataset

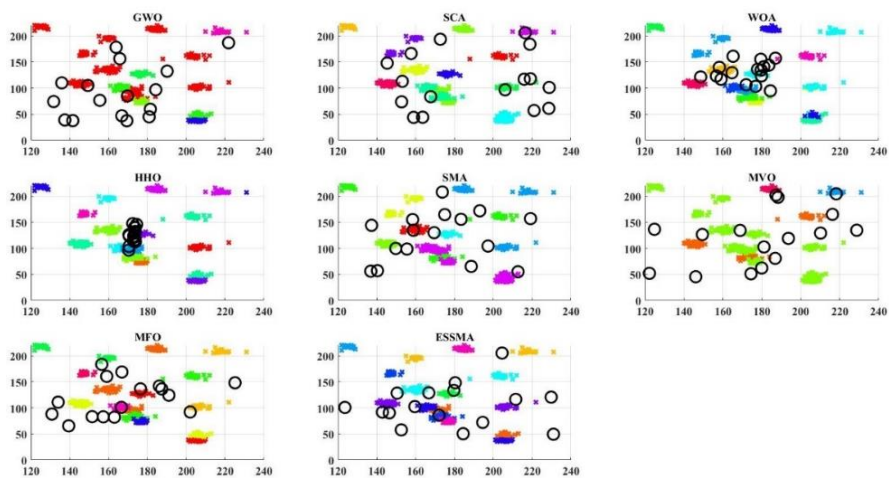


Fig. 30: Clustering results for the DIM64 dataset

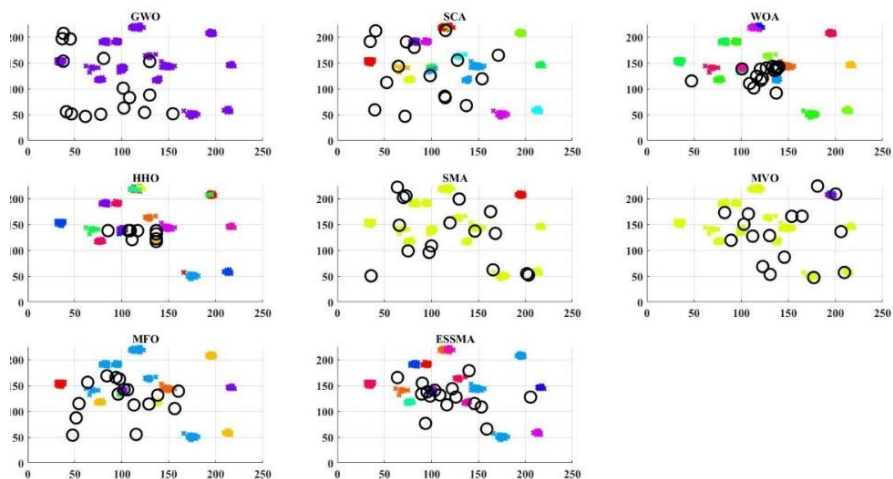


Fig. 31: Clustering results for the DIM128 dataset

where condition indicates that $S(i)$ ranks first half of the population, bF and wF represent respectively the best and the worst fitness, respectively, and r is a random variable in the interval $[0,1]$.

Food Wrapping

The slime mold evaluates the quality of food to decide how much time and energy (i.e., cytoplasmic flow) is needed to explore environments. If it estimates that the quality of the found food is poor, it retracts the path and changes position to explore other environments. Otherwise, slime mold increases the flow in a vein to exploit effectively the found food. Equation 7 expresses the flow of positive or negative information between the vein and slime mold. The condition simulates the adjustment of the search pattern by slime mold according to the quality of food. Based on this principle, the new location of slime mold is described by Eq. 9. Figure 3 illustrates the process of food quality evaluation:

$$\bar{X}^* = \begin{cases} rand.(UB - LB) + LB, & rand < z \\ \overline{Xb}(t) = \overline{vb} \cdot (W \cdot \overline{X_A}(t) - \overline{X_B}(t)), & r < p \\ \overline{vc} \cdot \overline{X}(t), & r \geq p \end{cases} \quad (9)$$

where UB and LB represent the Upper boundaries and lower boundaries in region search, respectively. z is a parameter that will be defined in the experimental implementation.

Oscillation

This phase concerns the wave sent from the outer legs of the slime mold inwards to change the flow of cytoplasm.

The variations of flow depend on the following variables; \overline{W} \overline{vb} \overline{vc} in addition to the width of the slime mold venous. W represents the frequency of oscillations in the slime mold algorithm that reflects the quality and concentration of food. The role of these oscillations is to help slime mold to speed up its approach to food. Moreover, the interaction of oscillation of \overline{vb} \overline{vc} and decides on the search mode; slime mold exploits intensively the region that has a good quality of food or explores more regions in the case of contacting food with poor quality. The main nutritional stage is Plasmodium of slime mold which linearly decays from one to zero.

Proposed Approach

The proposed approach described in Fig. 4 makes use of the eagle strategy to perform data clustering tasks. In the first step of ES, Levy flight random walks are used to explore efficiently and globally the search space and thus initialize the population of search agents with the resulting

solutions. These solutions belong to various regions of the search space and reflect the power of exploration of Levy's flight. Then, the solutions which obtain the best fitness values according to the clustering evaluation metric are recorded as promising solutions.

Afterward, the second phase starts and uses the slime mold algorithm as an efficient local optimizer to perform an intensive local search around the promising solutions. These two steps represent one round. The whole process is iterative and each round includes global exploration generating new populations in one or various regions, followed by another local search on those promising regions.

Algorithm 1: Pseudo algorithm for Eagle Strategy using Levy Flight and Slime Mould

Input: Dataset including N data objects with d dimensions and K number of clusters, number of rounds, the maximum number of iterations in each round
Output: Optimal positions of clusters representatives (i.e., centroids)

- 1 Initialize population size = 100, Round = 5, MaxIter = 50.
- 2 **while** $r \leq$ Round **do**
- 3 Global exploration using Levy flight and generation of initial solutions Evaluation of the fitness function and selection of a promising solution
- 4 **for** iter = 1 to MaxIter **do**
- 5 Generation of random solutions around this promising solution Intensive local search via Slime Mould Algorithm
- 6 **if** a better solution is found **then**
- 7 Update the current best solution
- 8 **else**

The proposed approach is population-based and considers P individuals which represent the set of solutions $P = [S_1, S_2, \dots, S_p]T$. The size of each solution is $K \times f$, such that K is the number of clusters and f is the number of features in a given dataset. The first step in the approach is to initialize randomly the candidate solutions in the boundary range:

$$S_i = lb + r \times (ub - lb) \quad (10)$$

where, S_i is the p th ($0 \leq i \leq P$) solution in the population, $r \in [0,1]$ is a random number, and lb and ub are the lower and upper bounds of the dataset taken under consideration.

The basic idea of ES is the initialization and running of an optimizer such as SMA. First, initialize the search space using Levy flight with fixed population size. Then,

SMA is used to carry out a search and find the best solutions after running for maximum iterations (MaxIter) and evaluation using 2. The best target value and its corresponding position vector are recorded for later use in a predefined number of rounds. In the next round, we generate a fraction of new solutions around the best solution, and then the rest (population size – fraction) solutions are generated in the whole search region. Then, SMA is run again for a maximum of iterations (MaxIter). Again, the new best solutions are saved and appended to the list of the best solution in the previous round. Then, we run exploration (LF) and exploitation (SAM) again. This mechanism is repeated until a maximum number of rounds is achieved. Algorithm 1 describes the different computation steps applied in the proposed approach.

Experimental Setup

To evaluate the performance of the proposed approach, we compared it to seven recently developed algorithms: Grey wolf Optimizer (GWO) Mirjalili *et al.* (2014), Sine Cosine Algorithm (SCA) Mirjalili (2016), whale optimization algorithm (WOA) Mirjalili and Lewis (2016), Harris Hawks Optimization (HHO) Heidari *et al.* (2019), slime mold algorithm (SMA) Li *et al.* (2020), Multi-Verse Optimizer (MVO) Mirjalili (2015) and Genetic Algorithm (GA) Katoch *et al.* (2021). The important parameters used are as follows:

- maximum number of rounds of ES-SMA = 5
- population size = 100
- number of iterations for all methods = 500
- several independent runs = 40.

Parameters of GA are set as follows: crossover type.

2-points, crossover probability: 0.8, mutation probability: 0.02. The other parameters of GWO, SCA, WOA, HHO, SMA, and MVO are set according to their corresponding references (Mirjalili *et al.*, 2014; Mirjalili, 2016; Mirjalili and Lewis, 2016; Heidari *et al.*, 2019; Li *et al.*, 2020; Mirjalili, 2015), respectively. These algorithms are run in the same machine for the same number of runs.

In the experiments, nine UCI datasets, and six shape datasets are considered for the evaluation of the method's performance in addition to three other high-dimensional datasets. The description of these datasets is given in Tables 1 and 2, respectively. We will not detail all the used datasets, but to get an idea of the data two dataset are explained. The Iris dataset is intended to be used for pattern recognition and it contains 3 classes referring to three types of iris plants where each class contains 50 instances. It also includes 4 features: Length and width of the sepal, and length and width of the petal. The second dataset is Breast Cancer data, which deals with the most

widespread cancers among women around the world, accounting for the majority of new cancer cases as well as cancer-related deaths. Features are computed from a digitized image of a fine needle aspirate of a breast mass. The set of features describes characteristics of the cell nuclei present in the image and includes diagnosis (malignant or benign), radius, texture, area, perimeter, smoothness, etc. All the considered algorithms were implemented using MATLAB R2017a and run on a machine with a 64-bit Windows 10 operating system, Core-i7 processor, and 8GB of RAM.

To further confirm the capability of the approach in working with high-dimensional problems, some of the considered datasets include more than 10 dimensions, namely CMC, Wine, Heart, and Path-based, and even better Dim-64 and Dim-128 include 64 and 128 dimensions, respectively. In addition, the birch dataset is too heavy datasets as they include a large number of objects ($N = 100000$) and has many clusters ($k = 100$).

Results and Discussion

This section reports and analyzes the experimental results obtained for the proposed approach and its comparison with the mentioned algorithms. We evaluated the performance of each algorithm using as metrics the average, standard deviation, best, and worst of the used objective, i.e., the sum of intra-cluster distances on every dataset and their convergence behavior. The values of the metrics are reported in Tables 3 and 4 for UCI and Shapes datasets. It is clear from these tables that our proposed approach ensures the minimum of the best results of the sum of intra-cluster distances obtained in the majority of the datasets. This shows that ESSMA can find high-quality clusters which have minimum dispersion and capture the cohesion feature in the representation space compared to the rest of the algorithms. Thus, it is an efficient alternative for the task of clustering.

Regarding the worst values of the sum of intra-cluster distances, results depicted in the same tables show that the results achieved by ESSMA are better than in other techniques on all the datasets, except for Compound, Spiral, and Yeast.

The average values of the sum of intra-cluster distances in all the experiments are reported in Table 3 and 4. Again, our approach outperforms all the considered algorithms. The obtained values for Standard Deviation (SD) are small and show the stability of ESSMA. In addition to these results, Fig. 15 - 31 visualize in two dimensions the obtained clustering for all datasets. Each cluster is distinguished by its unique color. In addition, the centroids of the clusters are illustrated by circles. Interestingly, the graphical analysis showed a distinguished group in all datasets.

To make the comparison clearer, we used two more measures based on the best-obtained values.

Table 1: UCI Datasets

Name	#Instances	#Features	#Classes
Iris	150	4	3
Seeds	210	7	3
Glass	214	9	6
Cancer	638	9	2
Wine	178	13	3
CMC	1473	10	3
Heart	270	13	2
Yeast	1484	8	10
Art	10218	8	10

Table 2: Shapes datasets

Name	#Instances	#Features	#Clusters
Flame	240	2	2
Jain	373	2	2
Path-based	178	13	3
Compound	399	2	6
Spiral	312	2	3
Birch1	100000	2	100
Sales transactions	811	53	4
DIM64	1024	64	16
DIM128	1024	128	16

Table 3: Best, worst, average, and standard deviation of WCSS obtained by the considered technique on the UCI datasets

Dataset name		Algorithms							
		GWO	SCA	WOA	HHO	SMA	MVO	GA	ESSMA
Iris	Best	96.66	129.75	97.22	96.81	96.66	96.66	96.66	96.66
	Worst	125.21	161.69	129.42	127.70	127.67	124.91	97.27	120.72
	Average	103.05	145.81	110.72	106.76	98.62	100.66	96.93	99.06
	Std	10.49	7.18	12.56	12.13	7.10	7.76	0.27	7.34
Seeds	Best	312.20	405.55	339.57	319.91	311.98	311.86	311.80	311.80
	Worst	318.09	497.59	444.15	430.09	313.27	359.32	311.82	311.81
	Average	312.92	457.66	379.17	364.97	312.48	322.66	311.80	311.80
	Std	1.06	23.90	28.95	27.33	0.29	12.88	0.00	0.00
Glass	Best	291.34	367.78	321.51	314.22	299.89	241.76	227.06	219.39
	Worst	367.29	492.36	443.30	478.19	395.29	324.13	264.25	259.36
	Average	326.06	426.85	382.97	391.32	349.05	283.96	253.99	245.57
	Std	19.06	31.70	31.29	41.15	31.84	22.45	8.75	10.96
Cancer	Best	2964.40	3211.63	2966.07	3107.95	2964.52	2965.74	2964.46	2964.39
	Worst	2964.53	3628.65	3003.09	4160.46	2965.41	3212.54	3076.66	2964.40
	Average	2964.43	3451.55	2978.31	3500.07	2964.75	3008.82	2983.39	2964.39
	Std	0.02	108.74	7.76	227.25	0.20	60.17	30.36	0.00
Wine	Best	16310.85	16644.85	16331.72	16332.61	16332.08	16293.54	16307.22	16292.41
	Worst	17030.88	17200.36	16527.40	16531.24	16436.58	16834.92	16828.95	16294.73
	Average	16359.72	16895.92	16392.47	16441.47	16385.01	16371.14	16441.91	16292.99
	Std	127.74	173.65	45.08	47.85	26.27	130.38	148.13	0.54
Cmc	Best	5729.82	6495.42	5957.67	5841.28	5739.37	5694.81	5693.74	5693.74
	Worst	6093.09	7816.15	7339.00	7301.02	5924.24	6253.58	5705.93	5693.94
	Average	5840.72	7000.55	6209.27	6246.07	5805.35	5876.36	5694.58	5693.83
	Std7	9.76	286.13	239.59	388.28	46.78	146.47	2.69	0.06
Heart	Best	10634.37	11203.01	10665.09	10782.00	10640.07	10627.54	10622.98	10623.05
	Worst	10707.40	12761.99	10997.36	13012.27	10723.47	10713.55	10853.36	10623.50
	Average	10659.85	12048.99	10774.42	11444.23	10677.05	10650.76	10671.69	10623.16
	Std	18.33	386.57	85.39	651.81	20.88	25.91	62.84	0.10
Yeast	Best	292.14	386.59	346.24	294.71	296.62	298.48	258.58	263.95
	Worst	378.48	566.71	385.00	377.11	380.13	377.18	280.90	330.79
	Average	341.39	451.40	368.54	325.14	357.103	45.42	270.58	299.47
	Std	24.85	51.93	11.42	18.34	24.42	22.07	6.44	17.46
Art	Best	513.91	554.28	514.24	513.97	513.90	513.90	513.90	513.90
	Worst	517.90	744.84	908.95	909.04	892.48	789.06	513.90	513.90
	Average	515.56	612.68	577.90	610.14	526.52	538.88	513.905	13.90
	Std	1.10	48.44	136.69	147.25	69.12	77.08	1.67E-4	7.28E-5

Table 4: Best, worst, average, and standard deviation of WCSS obtained by the considered algorithms on Shapes and high-dimensional datasets

Dataset name	Criteria	Algorithms							
		GWO	SCA	WOA	HHO	SMA	MVO	ESSMA	
Flame	Best	769.97	771.67	770.27	769.977	69.97	769.97	769.97	769.97
	Worst	769.97	784.63	774.96	792.270	769.97	769.97	769.97	769.97
	Average	769.97	775.84	771.85	775.070	769.97	769.97	769.97	769.97
	Std	1.72E-3	3.40	1.25	5.370	7.99E-4	2.18E-13	1.33E-5	
Jain	Best	2574.24	2577.16	2574.24	2574.240	2574.24	2574.00	24 2574.24	2574.24
	Worst	2574.26	2629.98	2580.67	2702.500	2574.24	2574.24	2574.24	2574.24
	Average	2574.25	2600.01	2574.80	2585.290	2574.24	2574.24	2574.24	2574.24
	Std	0.00	14.20	1.25	31.190	6.63E-4	9.44E-13	4.87E-5	7.80E-7
Compound	Best	1060.02	1205.45	1071.00	1107.960	1060.00	1059.97	1059.97	1059.97
	Worst	1111.11	1385.46	1370.75	1414.750	1196.09	1263.07	1195.95	1195.93
	Average	1074.93	1314.27	1177.72	1244.960	1096.27	1087.85	1113.05	1096.05
	Std	18.86	45.86	86.05	65.700	49.05	52.30	55.66	
Path-based	Best	1424.72	1450.04	1425.02	1428.400	1424.71	1424.71	1424.71	1424.71
	Worst	1425.16	1638.32	1576.00	1596.210	1425.14	1433.12	1503.86	1424.71
	Average	1424.75	1506.89	1455.64	1497.790	1424.73	1424.99	1433.77	1424.71
	Std	0.08	39.43	41.66	47.160	0.08	1.53	24.22	
Spiral	Best	1807.51	1819.03	1807.61	1807.550	1807.51	1807.51	1807.00	61 1807.51
	Worst	1808.43	1868.88	1827.38	1881.260	1808.17	1810.14	1821.48	1808.94
	Average	1807.66	1840.56	1813.46	1827.400	1807.60	1808.16	1809.68	1808.20
	Std	0.27	12.98	6.59	18.440	0.18	0.61	3.02	0.49
Birch1	Best	3.56E+09	4.55E+09	3.94E+09	4.15E+09	3.52E+09	3.47E+09	4.09E+09	3.24E+09
	Worst	4.70E+09	4.71E+09	4.15E+09	4.57E+09	3.68E+09	3.66E+09	4.55E+09	3.49E+09
	Average	4.24E+09	4.65E+09	4.06E+09	4.32E+09	3.58E+09	3.56E+09	4.26E+09	3.37E+09
	Std	5.26E+08	5.36E+07	7.37E+07	1.10E+08	4.20E+07	5.51E+07	1.43E+08	8.35E+07
Sales Transactions	Best	29783.32	51258.32	41469.95	47852.290	30979.91	27543.07	94607.63.00	20633.85
	Worst	33423.49	52310.79	47337.14	48792.790	32485.72	20144.25	99079.05	21027.92
	Average	31856.24	52013.77	44089.91	48358.700	31525.04	30195.03	95969.82	20855.29
	Std	1329.11	427.98	2565.88	387.860	565.87	1698.8	1803.04	159.51
DIM64	Best	397832.75	511475.49	417731.35	419839.380	376037.88	398900.17	369356.92	362892.86
	Worst	407937.01	519955.28	419426.31	427408.820	424662.65	414515.26	386077.25	376270.73
	Average	403887.80	515796.34	418825.01	423270.910	405079.88	409303.04	378218.82	368540.68
	Std	5342.43	4242.21	948.69	3833.840	25655.40	9009.15	8405.21	6927.73
DIM128	Best	613177.84	747073.69	607685.26	607223.280	612483.78	610427.27	598844.85	570999.99
	Worst	631671.37	767435.81	609736.28	609704.890	615322.70	612918.21	606886.22	580285.69
	Average	620612.92	758252.76	608680.80	608515.260	613790.66	611871.76	601676.70	574478.37
	Std	9764.69	10326.76	1026.82	1243.970	1432.79	1292.29	4517.28	5062.01

Table 5: Resulting values of the Wilcoxon test for a statistically significant level of $\alpha = 0.05$

ESSMA vs	GWO	SCA	WOA	HHO	SMA	MVO	GA
Flame	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	2.58E-11	1.79E-4
Jain	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	2.83E-11	6.79E-08
Compound	<u>0.05</u>	3.02E-11	7.74E-06	5.57E-10	1.84E-02	1.91E-02	43.8E-3
Path-based	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	5.39E-10	1.8E-05
Spiral	1.68E-03	3.02E-11	1.39E-06	4.20E-10	6.91E-04	<u>0.46</u>	33.71E-3
Birch 1	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04	3.30E-04	1.83E-04
Sales Transactions	1.51E-01	7.94E-03	7.94E-03	7.94E-03	4.21E-01	7.94E-03	7.94E-03
DIM64	1.00E-01	1.00E-01	1.00E-01	1.00E-01	3.00E-01	1.00E-01	2.00E-01
DIM128	1.00E-01	1.10E-01	1.50E-01	2.00E-01	1.00E-01	1.00E-01	1.00E-01

Table 6: Resulting values of the Wilcoxon test for the statistically significant level at $\alpha = 0.05$

ESSMA vs	GWO	SCA	WOA	HHO	SMA	MVO	GA
Iris	3.35E-08	3.02E-11	1.10E-08	1.85E-08	1.07E-07	<u>0.07</u>	5.29E-04
Seeds	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	6.25E-04
Glass	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	1.17E-09	2.22E-04
Cancer	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	6.79E-08
Wine	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	6.79E-08
CMC	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	<u>0.69</u>
Heart	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	1.59E-05
Yeast	2.83E-08	3.02E-11	3.02E-11	4.74E-06	3.82E-10	7.77E-09	9.17E-08
Art	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.34E-11	1.04E-07	1.79E-04

Table 7: Average of rankings over all datasets for the considered algorithms

Algorithm	Convergence
GWO	3.36
SCA	8.00
WOA	5.63
HHO	5.93
SMA	3.14
MVO	2.32
GA	3.13
ESSMA	1.16

Table 8: Average of the best-obtained values over all datasets for the considered algorithms

Algorithm	Convergence
GWO	50.03E+15
SCA	81.72E+15
WOA	57.90E+15
HHO	75.34E+15
SMA	51.88E+15
MVO	50.09E+15
GA	58.26E+15
ESSMA	49.956E+15

We first calculate the average of the best values over all the datasets as reported in Table 8. The average of the best values obtained using our method is the smallest, followed by GWO and MVO. Next, we ranked the algorithms based on their best-obtained values and averaged those rankings. The average of the ranks across all datasets is reported in Table 7 and again our approach ranks first, followed by MVO and GA.

Statistical Analysis

To perform an accurate evaluation process, a non-parametric statistical test is used on the experimental results. The Wilcoxon test is conducted at a 5% level of significance. The null hypothesis is rejected if the p-value < 0.05. Table 5 contains the obtained p-value of ESSMA compared to the chosen meta-heuristic with the best performance illustrated in bold text. It can be seen from Table 5 and 6 that ESSMA outperforms SCA, WOA, and HHO in all used datasets. Also, ESSMA outperforms generally the GWO (resp. SMA) except in the datasets Compound; besides, it does not outperform GA only in CMC. Using this statistical test, it turns out that MVO is the competitor of our approach since the p-value obtained in one dataset, namely the Spiral datasets is higher than the significance level.

Convergence Analysis

Figures 5 to 14 show the convergence behavior of the algorithms graphically for all the considered datasets. The vertical axis represents the fitness values obtained in each iteration lying in the horizontal axis while the horizontal axis refers to the number of iterations. ESSMA has shown the best behavior in all datasets except the convergence curve applied to the Yeast dataset. In addition, the curves demonstrate a rapid switch from exploration to exploitation. Thus, the ESSMA proves an efficient performance compared to the well-known algorithm.

Statistically speaking, the ESSMA algorithm is more successful in avoiding the local optima and efficiently evades the local traps in all datasets. According to the mathematical formulation of the ESSMA algorithm, at each iteration, a fraction of the solutions is replaced by other ones by using the so-called Levy flight which is a kind of random walk that forces the search individuals or particles to perform random steps towards or outwards the best agent. This promotes exploration of the search space which leads to finding diverse structures during optimization. Besides, the position update of the rest of the portion of the population is ensured by using the standard SMA to intensify the exploitation phase. Per this study, it is recommended to use the ESSMA metaheuristic as an efficient optimization algorithm, particularly for hybrid applications such as data clustering. Most importantly, due to its high exploratory capabilities that prevent getting stuck in a local optimum when optimizing the distance objectives. Additionally, its high exploitative property explains why the ESSMA-based clustering can converge efficiently towards the global optimum in complex search space. The results of this study show that although evolutionary algorithms have high exploration and can achieve high accuracy in determining the centroid positions, as shown by the use of ground truth data, the problem of data clustering requires intelligently avoiding local optima throughout the optimization process. The achieved results show that the ESSMA is very efficient in this regard.

It is worth mentioning that ESSMA is strongly recommended when the considered problem and dataset are complex and multi-dimensional with a large number of features. Otherwise, it is more convenient to use gradient-based training algorithms where the dataset is small and includes very few features as it will be faster and less computationally expensive.

Conclusion

This study has investigated the application of the Eagle Strategy on Levy Flight and SMA to solve the clustering problems. Results show that the proposed approach is capable of searching for the best cluster centroids and ensuring the avoidance of the local minima traps, which is the main disadvantage of many approaches such as the K-means Algorithm. The clustering problem is formulated as an optimization problem that aims to minimize the sum of intra-cluster distances. ESSMA exploits the exploration power of Levy Flight and the exploitation afforded by SMA. The performance of the method has been evaluated based on eighteen datasets from UCI and Shapes repositories and compared with seven well-known algorithms GWO, SCA, WOA, HHO, SMA, MVO, and GA. The empirical results showed that the introduced ESSMA Clustering method achieves higher performance than other algorithms on almost all the datasets.

Researchers in biology are still trying to understand comprehensively the foraging behaviors of different bacteria. Their results can inspire researchers to update the Slim Mould algorithm. In addition, future work must develop how to use the Slim Mould Algorithm in hybrid mode to enhance the exploration phase. In addition, the enhancement of a Fuzzy algorithm based on ESSMA to cluster data taking into account the uncertain data, the noisy data, and the unbalanced clusters can be an interesting research track.

Acknowledgment

The author would like to thank the responsible reviewers and editors for their constructive feedback.

Funding Information

The research presented in this manuscript did not receive any external funding.

Ethics

The authors declare that the paper does not involve any ethical issues and it presents their own research in a truthful manner without any influence from a competing financial interests or personal relationships.

Author's Contributions

Rachid Oucheikh: Conceptualization, formal analyses, development, original draft and Investigation.

Achraf Touil: Conceptualization, review, and editing.

Mouhsene Fri: Review, methodology, publication.

References

- Abualigah, L., Gandomi, A. H., Elaziz, M. A., Hussien, A. G., Khasawneh, A. M., Alshinwan, M., & Houssein, E. H. (2020). Nature-inspired optimization algorithms for text document clustering—a comprehensive analysis. *Algorithms*, 13(12), 345. <https://doi.org/10.3390/a13120345>
- Adam, S. P., Alexandropoulos, S. A. N., Pardalos, P. M., & Vrahatis, M. N. (2019). No free lunch theorem: A review. *Approximation and Optimization*, 57-82. <https://doi.org/10.1109/ACCESS.2021.3094471>
- Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1), 243-256. <https://doi.org/10.1016/j.patcog.2012.07.021>
- Bagirov, A. M., Karmitza, N., and Taheri, S. (2020). Metaheuristic clustering algorithms. In *Unsupervised and Semi-Supervised Learning*, pages 165–183. Springer International Publishing. https://doi.org/10.1007/978-3-030-37826-4_6
- Borah, S., & Ghose, M. K. (2009). Performance analysis of AIM-K-means & K-means in quality cluster generation. *arXiv preprint arXiv:0912.3983*. <https://doi.org/10.48550/arXiv.0912.3983>
- Chechkin, A. V., Metzler, R., Klafter, J., & Gonchar, V. Y. (2008). Anomalous transport: Foundations and applications. *Introd Theor Lévy Flights*.
- Chen, Z., & Liu, W. (2020). An efficient parameter adaptive support vector regression using K-means clustering and chaotic slime mold algorithm. *IEEE Access*, 8, 156851-156862. <https://doi.org/10.1109/ACCESS.2020.3018866>
- Esmín, A. A., Coelho, R. A., & Matwin, S. (2015). A review of particle swarm optimization algorithm and its variants to clustering high-dimensional data. *Artificial Intelligence Review*, 44(1), 23-45. <https://doi.org/10.1007/s10462-013-9400-4>
- Ezugwu, A. E. (2020). Nature-inspired metaheuristic techniques for automatic clustering: a survey and performance study. *SN Applied Sciences*, 2(2), 1-57. <https://doi.org/10.1007/s42452-020-2073-0>
- Hämäläinen, J., Jauhiainen, S., & Kärkkäinen, T. (2017). Comparison of internal clustering validation indices for prototype-based clustering. *Algorithms*, 10(3), 105. <https://doi.org/10.3390/a10030105>
- Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 97, 849-872. <https://doi.org/10.1016/j.future.2019.02.028>

- Janani, R., & Vijayarani, S. (2019). Text document clustering using spectral clustering algorithm with particle swarm optimization. *Expert Systems with Applications*, 134, 192-200.
<https://doi.org/10.1016/j.eswa.2019.05.030>
- Karaboga, D., & Ozturk, C. (2011). A novel clustering approach: Artificial Bee Colony (ABC) algorithm. *Applied Soft Computing*, 11(1), 652-657.
<https://doi.org/10.1016/j.asoc.2009.12.025>
- Kareiva, P., & Odell, G. (1987). Swarms of predators exhibit "prey-taxis" if individual predators use area-restricted search. *The American Naturalist*, 130(2), 233-270. <https://doi.org/10.1086/284707>
- Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review of the genetic algorithm: Past, present, and future. *Multimedia Tools and Applications*, 80(5), 8091-8126.
<https://doi.org/10.1007/s11042-020-10139-6>
- Li, K., Torres, C. E., Thomas, K., Rossi, L. F., & Shen, C. C. (2011). Slime mold-inspired routing protocols for wireless sensor networks. *Swarm Intelligence*, 5(3), 183-223.
<https://doi.org/10.1007/s11721-011-0063-y>
- Li, S., Chen, H., Wang, M., Heidari, A. A., & Mirjalili, S. (2020). Slime mold algorithm: A new method for stochastic optimization. *Future Generation Computer Systems*, 111, 300-323.
<https://doi.org/10.1016/j.future.2020.03.055>
- Maulik, U., & Bandyopadhyay, S. (2000). Genetic algorithm-based clustering technique. *Pattern Recognition*, 33(9), 1455-1465.
[https://doi.org/10.1016/S0031-3203\(99\)00137-5](https://doi.org/10.1016/S0031-3203(99)00137-5)
- Maulik, U., & Saha, I. (2010). Automatic fuzzy clustering using modified differential evolution for image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 48(9), 3503-3510.
<https://doi.org/10.1109/TGRS.2010.2047020>
- Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, 89, 228-249.
<https://doi.org/10.1016/j.knsys.2015.07.006>
- Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-based systems*, 96, 120-133.
<https://doi.org/10.1016/j.knsys.2015.12.022>
- Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, 95, 51-67.
<https://doi.org/10.1016/j.advengsoft.2016.01.008>
- Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Computing and Applications*, 27(2), 495-513.
<https://doi.org/10.1007/s00521-015-1870-7>
- Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69, 46-61.
<https://doi.org/10.1016/j.advengsoft.2013.12.007>
- Mostafaie, T., Khiyabani, F. M., & Navimipour, N. J. (2020). A systematic study on meta-heuristic approaches for solving the graph coloring problem. *Computers & Operations Research*, 120, 104850.
<https://doi.org/10.1016/j.cor.2019.104850>
- Nagurney, A. (2010). Supply chain network design under profit maximization and oligopolistic competition. *Transportation Research Part E: Logistics and Transportation Review*, 46(3), 281-294.
<https://doi.org/10.1016/j.tre.2009.11.002>
- Nakagaki, T., Yamada, H., & Ueda, T. (2000). Interaction between cell shape and contraction pattern in the *Physarum plasmodium*. *Biophysical chemistry*, 84(3), 195-204. [https://doi.org/10.1016/S0301-4622\(00\)00108-3](https://doi.org/10.1016/S0301-4622(00)00108-3)
- Nanda, S. J., & Panda, G. (2014). A survey on nature-inspired metaheuristic algorithms for partitioned clustering. *Swarm and Evolutionary Computation*, 16, 1-18. <https://doi.org/10.1016/j.swevo.2013.11.003>
- Nayyar, A., Le, D. N., & Nguyen, N. G. (Eds.). (2018). *Advances in swarm intelligence for optimizing problems in computer science*. CRC press.
- Nayyar, A., & Nguyen, N. G. (2018). Introduction to swarm intelligence. In *Advances in Swarm Intelligence for Optimizing Problems in Computer Science*, pp. 53-78. Chapman and Hall/CRC.
- Nemmich, M. A., Debbat, F., & Slimane, M. (2018, April). A data clustering approach using a bees algorithm with a memory scheme. In *International Conference on Computer Science and its Applications* (pp. 261-270). Springer, Cham. <https://doi.org/10.4018/IJOCI.2018100102>
- Saemi, B., Hosseinabadi, A. A. R., Kardgar, M., Balas, V. E., & Ebadi, H. (2016, August). Nature-inspired partitioning clustering algorithms: A review and analysis. In *International workshop soft computing applications* (pp. 96-116). Springer, Cham. https://doi.org/10.1007/978-3-030-52190-5_4
- Sheikholeslami, G., Chatterjee, S., & Zhang, A. (2000). WaveCluster: A wavelet-based clustering approach for spatial data in very large databases. *The VLDB Journal*, 8(3), 289-304.
<https://doi.org/10.1007/s007780050009>
- Shi, Z., & Pun-Cheng, L. S. (2019). Spatiotemporal data clustering: A survey of methods. *ISPRS International Journal of Geo-Information*, 8(3), 112. <https://doi.org/10.3390/ijgi8030112>
- Singh, T. (2020). A chaotic sequence-guided Harris hawks optimizer for data clustering. *Neural Computing and Applications*, 32(23), 17789-17803.
<https://doi.org/10.1007/s00521-020-04951-2>

- Srinivasu, P. N., Ahmed, S., Alhumam, A., Kumar, A. B., & Ijaz, M. F. (2021). An AW-HARIS-based automated segmentation of human liver using CT images. *Comput Mater Contin*, 69(3), 3303-3319.
- Tan, S. C., Ting, K. M., & Teng, S. W. (2011). Simplifying and improving ant-based clustering. *Procedia Computer Science*, 4, 46-55. <https://doi.org/10.1016/j.procs.2011.04.006>
- Tripathi, A. K., Sharma, K., & Bala, M. (2018). A novel clustering method using enhanced grey wolf optimizer and MapReduce. *Big data research*, 14, 93-100. <https://doi.org/10.1016/j.bdr.2018.05.002>
- Wang, T., Ren, C., Luo, Y., & Tian, J. (2019). NS-DBSCAN: A density-based clustering algorithm in network space. *ISPRS International Journal of Geo-Information*, 8(5), 218. <https://doi.org/10.3390/ijgi8050218>
- Yan, X., Zhu, Y., Zou, W., & Wang, L. (2012). A new approach for data clustering using a hybrid artificial bee colony algorithm. *Neurocomputing*, 97, 241-250. <https://doi.org/10.1016/j.neucom.2012.04.025>
- Yang, H., Mayne, R., & Deng, Y. (2019). A Bio-inspired Network Design Method for Intelligent Transportation. *International Journal of Unconventional Computing*, 14.
- Yang, X. S., & Deb, S. (2010). Eagle strategy using Lévy walk and firefly algorithms for stochastic optimization. In Nature inspired cooperative strategies for optimization (NICSO 2010) (pp. 101-111). *Springer, Berlin, Heidelberg*. <https://doi.org/10.1007/s11069-021-05165-y>
- Zhang, W., & Di, Y. (2020). Model-based clustering with measurement or estimation errors. *Genes*, 11(2), 185. <https://doi.org/10.3390/genes11020185>
- Zhang, X., Adamatzky, A., Yang, X. S., Yang, H., Mahadevan, S., & Deng, Y. (2016). A Physarum-inspired approach to supply chain network design. *Science China Information Sciences*, 59(5), 1-14. <https://doi.org/10.1007/s11432-015-5417-4>