MATHEMATICS AND COMPUTATIONAL SCIENCES

Vol 6(1), 2025, pp:116-146

DOI:10.30511/mcs.2025.2041731.1235

ISSN:2717-2708

http://mcs.qut.ac.ir/

Qom University of Technology

Efficient cluster center optimization: A novel hybrid metaheuristic

Sæideh Barkhordari Firozabadi [1], **Seyed Abolfazl Shahzadeh Fazeli** [2], **Jamal Zarepour Ahmadabadi** [3], **Seyed Mehdi Karbassi** [4]

[1,2,3]Department of Computer Science, Yazd University, Yazd, Ira

[4]Department of Applied Mathematics, Yazd University, Yazd, Iran

Abstract: Metaheuristics have proved highly effective in addressing optimization challenges. Various algorithms address the clustering problem to find optimal centers for the clusters. One of the disadvantages of some of these algorithms is stagnation in local optima, especially for big data. If this problem is not properly solved, the clustering process will suffer. This research introduces a new hybrid method by merging the capabilities of two metaheuristic algorithms: Harris hawks optimization algorithm (HHO) and slime mould algorithm (SMA). These metaheuristic methods are employed to determine the best location for the cluster centers. Optimization aims to reduce intra-cluster distance. In other words, the data points of each cluster should be close to its cluster center and also to avoid local optima. The effectiveness of these techniques is assessed and contrasted with the SMA and HHO algorithms on Iris, Vowel and Wine data sets. Compared to mentioned algorithms, our proposed method exhibits significantly improved convergence speed. The results also proved this method can properly find the optimal centers for clustering which finally improves the performance of the proposed method.

Keywords: Clustering; Metaheuristic; Slime mould algorithm; Harris hawks optimization algorithm;
2010 Mathematics Subject Classification: 68T10; 62H30
Receive: 24 September 2024, **Accepted:** 31 January 2025

1 Introduction

One of the key techniques for knowledge discovery in databases is clustering [33]. According to the data and problem conditions, different algorithms are used for clustering such as k-means [19, 9], k-medoids [9], Fuzzy C-Means (FCM) [3, 9], DBSCAN [5, 25], their combinations [2], and etc. One of the important issues in clustering is finding the optimal centers. The identification of optimal centers for the clusters, is a crucial task in various domains such as data analysis, pattern recognition, and machine learning as it enables data segmentation, pattern discovery, classification, prediction, and understanding of underlying structures. Also use of the optimal centers instead of random initial centers accelerates the convergence for finding more accurate solutions. According to studies, metaheuristic algorithms have many capabilities in different issues. The creation of nature-inspired metaheuristic algorithms have exploded in recent years as a means of addressing challenging optimization issues that cannot be efficiently solved

[1]s.barkhordari@stu.yazd.ac.ir
[2]Corresponding author: fazeli@yazd.ac.ir
[3]zarepourjamal@yazd.ac.ir
[4]smkarbassi@yazd.ac.ir

using conventional techniques. These algorithms do not rely on gradient information and instead repeatedly call the objective function to identify the global optimum. By narrowing down the search space, these algorithms effectively find solutions. Leveraging the intelligence of natural phenomena, metaheuristic algorithms are capable of discovering near-global or global solutions [6, 16]. Recent decades have witnessed the introduction of various metaheuristic algorithms inspired by natural processes, physics, collective behavior, art, or mathematical principles of classical approaches for instance, metaheuristic algorithms like Ant Colony Optimization (ACO) [26], Genetic Algorithm (GA) [12], Grasshopper Optimization Algorithm (GOA) [30], Differential Evolution (DE) [26], Cuckoo Search (CS) [7], Particle Swarm Optimization(PSO) [15, 11], Grey Wolf Optimizer (GWO) [21], Mountain Gazelle Optimizer (MGO) [1], Stochastic Paint Optimizer (SPO) [14], Golden Eagle Optimizer (GEO) [22] and Tunicate Swarm Algorithm (TSA) [13]. Different problems can be solved by combining metaheuristic algorithms [28]. Many researchers utilized these algorithms in combination such as hybrid PSOGSA Algorithm [20], MGO-PSO Algorithm [23] and MFO-PSO Algorithm [24]. Metaheuristic algorithms can be combined with other algorithms to solve different problems. Combining these algorithms with clustering algorithms will lead to satisfactory results. Metaheuristic algorithms have been successful in reducing stagnation in local optima and using their strengths can be very helpful. Sunita et al. in [31] introduced hybrid PSO plus KM and offered a comparison of the three clustering techniques: KM, PSO, and a hybrid approach combining PSO and KM. The results showed that combined PSO plus KM performs better than both other methods. In [27], GWO-GOA is proposed. The approach, which combined the GWO-GOA and FCM clustering techniques, was employed to cluster the chosen optima, employed eight data sets and performed better than previous algorithms in terms of precision, specificity, sensitivity, f-measure, and recall. In [10], authors proposed a novel dynamic and incremental method for document clustering called CS-LSI. It is based on the most recent optimization of the Cuckoo Search (CS). By using a newly provided index, the proposed approach automatically calculates the number of clusters and obtains better results. To summarize, the key findings of this study are:

1. A hybrid approach was proposed that combines computational intelligence techniques to find optimal centers for clusters.
2. The method was derived from SMA and HHO based on clustering algorithms, both of which have shown promising results in optimizing the placement and distribution of data points within clusters.
3. The combination of different approaches enhances performance, accuracy, and robustness in solving complex problems during the algorithm.
4. Experimental results and convergence curves were compared with the several methods.
5. The proposed method is evaluated on three datasets: Iris, Wine, and Vowel.
6. In terms of both computing time and solution quality, the proposed technique performs better than the mentioned methods.
7. The results affirm the potential of metaheuristic algorithms to revolutionize the field of clustering which pave new paths for solving complex optimization problems.

This paper is structured as follows: a detailed explanation of the metaheuristic algorithm and techniques applied is provided in Section 2. Section 3 offers the proposed method. A detailed analysis of the experimental observations are given in Section 4. Section 5 includes a summation of the key findings.

2 Metaheuristic algorithm

Two categories of optimization algorithms exist: approximate algorithms and exact algorithms. Optimal solution can be found exactly by exact algorithms, however, when handling challenging the complexity of optimization problems, their execution time is proportional to the dimensions of the problems, which increases exponentially that are not efficient enough in such problems.
Approximate algorithms care design of provide a near-optimal solution within a reasonable amount of time. This is especially beneficial for complex optimization problems. The three types of approximate algorithms are super

heuristic, metaheuristic, and heuristic two primary issues of heuristic algorithms are their premature convergence to these points and stagnation in local optimal points, but metaheuristic algorithms have a good performance in dealing with these problems [17]. In fact, metaheuristic algorithms can find good solutions for hard problems in a short time and provide appropriate solutions when dealing with local optimal points. Metaheuristic algorithms, by using two important and key mechanisms called exploration and exploitation, possess the capacity to transcend the local optimum and arrive at the global optimum.
Metaheuristic algorithms can be categorized using a range of criteria:

1. One answer based and population based: In population based algorithms, the search process considers a population of answers, while in algorithms based on one answer, only one answer is changed in the search.
2. Inspired by nature optimization techniques and not inspired by nature: The behavior of living things and the natural world are major sources of inspiration for many metaheuristic algorithms, among which some metaheuristic algorithms are not the same.
3. With memory and without memory: Some metaheuristic algorithms use memory and store it in themselves (e.g. simulated refrigeration). This is despite the fact that certain metaheuristic algorithms do not employ the data that is obtained throughout the search phase since they are memoryless.
4. Deterministic and probabilistic: In probabilistic metaheuristic algorithms, they solve the problem by using a set of probabilistic rules which guide the search process, but a metaheuristic algorithm deterministic, such as forbidden search, makes a decision to solve the problem in a deterministic way [35].

Various types of algorithms in recent decades have been developed that is applicable to a diverse range of problems. The following provides an explanation of two examples: SMA and HHO.

2.1 Slime mould algorithm

SMA is a metaheuristic and powerful population-based optimizer method created by Shimin et al. [32] in 2020. This algorithm draws inspiration from the intelligent actions of single-celled life, under the name of mucous mould. Because of its straightforward structure and acceptable convergence, this approach has drawn the interest of numerous scholars. The single-celled organisms in this mould, despite being deprived of the power of thinking, can intelligently find their way through winding paths. When the slime mould was searching for food, to mimic the positive and negative feedback produced by it, SMA uses weights.
A eukaryote that lives in frigid, and damp environments is slime mould. Slime mould can even grow if there is enough food in the environment. A propagating wave is produced by the bio-oscillator when it gets close to a food source. If the conditions are not suitable, the slime mould is capable of creating optimal paths toward regions with higher concentrations of food so they receive the highest possible concentration of nutrients. Slime mould possesses the ability to select the food supply based on concentration, especially when presented with options of varying quality. The search patterns of the slime template are dynamically adjusted according to the food quality. In fact the slime mould depending on the conditions (for example, when it encounters high quality food) must be able to determine when to search in another region and when to leave this one [32]. In the following, some characteristics of slime mould and mechanism is mathematically modeled:

1. Approach food:

   To simulate the slime mould's behavior, the following rules are listed. When it approached the food source as a mathematical equation:

$$X_{t+1} = \begin{cases} X_b(t) + v_b.\,(W.X_A(t) - X_B(t)) & r < p \\ v_c.X_t & r \geq p. \end{cases} \tag{2.1}$$

   In this case, t denotes the present step, X denotes the slime mould location, X_A and X_B indicate a random pair of individuals, and X_b represents the slime mould's location with the highest odor concentration that is

now detected. v_b is selected in $[-a, a]$, v_c follows a linear decreasing trend during iterations. Firstly, it is one and finally becomes zero. The slime mould's weight is indicated by W. The value of p is acquired in this way:

$$p = \tanh|S(i) - DF|. \tag{2.2}$$

The fitness of X, for $i \in 1, 2, \cdots, n$ is indicated with $S(i)$, DF indicates the optimal fitness attained over iterations.

v_b is obtained by:

$$v_b = [-a, a] \tag{2.3}$$

where

$$a = \tanh^{-1}\left(-\left(\frac{t}{max_t}\right) + 1\right). \tag{2.4}$$

The weight of slime mould is obtained by the following procedure:

$$W(SmellIndex(i)) = \begin{cases} 1 + r\log((b_F - S(i))/(b_F - w_F) + 1) & condition \\ 1 - r\log((b_F - S(i))/(b_F - w_F) + 1) & others \end{cases} \tag{2.5}$$

$$SmellIndex = sort(S). \tag{2.6}$$

Here r is selected randomly within [0,1], condition $S(i)$ ranks the upper 50% of the population and the best fitness achieved throughout the present iterative phase is represented by b_F.

The $SmellIndex$ sorts the order in which the fitness values (in the minimization problem, ascends). The poorest fitness value currently found in the iterative procedure is represented with w_F.

2. Wrap food:

   The mathematical formula for updating the slime mould's location, is:

$$X^* = \begin{cases} rand(UB - LB) + LB & rand < z \\ X_b(t) + v_b(WX_A(t) - X_B(t)) & r < p \\ v_c X(t) & r \geq p. \end{cases} \tag{2.7}$$

   r and rand denote the random value in [0, 1], the search range lower and upper bounds are denoted by the terms LB and UB, respectively.

3. Oscillation:

   At the end of the procedure, v_b tends to zero in value and oscillates randomly between $[-a, a]$. When the quantity of iterations increases, v_c eventually approaches zero and fluctuates between [-1, 1].

Algorithm 1 provides a description of the SMA algorithm.

2.2 Harris hawks optimization

Heidari et al. [8] devised HHO in 2019, based on population gradient-free optimization technique that draws inspiration from Harris hawk hunting tactics. This algorithm mimics the social interaction and strategic hunting patterns of these birds to solve complex problems. The concept of HHO is rooted in the natural world, drawing its surprise pounce and core principles from the collaborative hunting tactics of wild Harris hawks. This tactic involves multiple hawks working together to jointly attack on a rabbit-like prey from various angles in an effort to startle it, then simultaneously converge on a discovered fleeing rabbit [8].

Algorithm 1 SMA algorithm [32]

Inputs: Population size (N), maximum number of iterations (max_t)
Outputs: Optimal solution

1. Set the slime mould positions $X_i(i = 1, 2, \ldots, n)$ initially
2. **While**(The stopping requirement is not satisfied) **do**
   (a) Compute the fitness of each slime mould
   (b) Compute the W as in Eq. (2.5)
   (c) **for**(all search portion (X_i)) **do**
      i. Update p, v_b, v_c
      ii. Update positions as in Eq. (2.7)
   end for
   end while
3. **Return** best fitness and the corresponding optimal solution X_b

1. Exploration phase:
   With their keen vision, Harris hawks are able to track and identify their prey. They wait, watch, and scan the desert area for signs of prey, which may take many hours to find because sometimes it is difficult to observe the prey. The Harris hawks use two different ways to detect prey while perching in random sites. To each perching strategy, given a chance equal to q. They randomly choose a tree to perch among tall trees, which is modeled for condition in Eq. (2.8):

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1 \left| X_{rand}(t) - 2r_2 X(t) \right| & q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) & q < 0.5. \end{cases} \tag{2.8}$$

   In this case, in the $t+1$ iteration, the hawks' position vector is $X(t+1)$, the rabbit's position is $X_{rabbit}(t)$, and the current position vector of the hawks is $X(t)$. A hawk is picked at random from the extant population and $X_{rand}(t)$ denotes it, r_1, r_2, r_3, r_4, and q are selected randomly from (0,1) and modified following each iteration. The present hawk population's average location is X_m. To produce random values inside the home range of the group defined by the LB and UB , a model is suggested.
   By using Eq. (2.9) the average position of hawks is gained:

$$X_m(t) = \frac{1}{N} \sum_{i=1}^{N} X_i(t). \tag{2.9}$$

   Here N denotes the entire hawk count and $X_i(t)$ indicates the location of each hawk. The simplest rule is utilized but there are various methods available to determine the average location [8].
2. Transition from exploration to exploitation:
   Switch between exploration and exploitation can be done dynamically in the algorithm. A prey's escaping energy is denoted by E that symbolized by the equation that follows:

$$E = 2E_0(1 - \frac{t}{T}). \tag{2.10}$$

Here E_0 is the initial state of E. In during the iterations, there is a trend toward a diminishing dynamic escape energy E. When $|E| \geq$1, exploration takes place; in subsequent phases, exploitation occurs when $|E| <$1 [8].

3. Exploitation phase:
   During this stage, many chasing techniques are used in actual settings. The rabbit constantly tries to get away from the hunter. Assume that the probability of a rabbit is r. The prey escapes if $r <$0.5, otherwise it is caught. Depending on the rabbit's energy level, the hawks will surround it in various directions and with varying degrees of softness or force. To catch the rabbit, they will then execute a hard or soft besiege. In real situations, hawks intensify the process of encirclement and slowly get closer to the target prey. Then, by making a surprise throw, they try to catch the tired prey with more luck, because the prey loses more energy after running away for a few minutes. The parameter E is used for mathematical modeling of this strategy. To mimic the attack phase, the HHO algorithm incorporates four different ways, that is explained in the following [8].

   (a) Soft besiege:
   When both $r \geq 0.5$ and $|E| \geq 0.5$, the rabbit attempts to escape using its remaining energy by random misguided jumps. Harris hawks slowly surround it to further tire the rabbit. Finally, the rabbit cannot continue to run away and they catch the prey with a surprising jump. These rules determine this behavior:

$$X(t+1) = \Delta X(t) - E\,|JX_{rabbit}(t) - X(t)| \tag{2.11}$$

$$\Delta X(t) = X_{rabbit}(t) - X(t). \tag{2.12}$$

   The disparity in location between the rabbit's position vector and the current iteration is denoted by $\Delta X(t)$. Additionally, r_5 is selected randomly on (0,1). $J = 2(1 - r_5)$ represents the rabbit's random jumping capability, which varies randomly in all stages of escape during iteration.

   (b) Hard besiege:
   When both $r \geq$0.5 and $|E| <$0.5, the Harris hawk surrounds the target prey with great effort, the rabbit has a runaway energy that is low and it is very tired. Finally, they catch the prey with a spectacular jump. The Eq. (2.13) are used to update the positions [8]:

$$X(t+1) = X_{rabbit}(t) - E\,|\Delta X(t)|\,. \tag{2.13}$$

   (c) Soft encirclement with escalating, fast dives:
   This method is smarter than the previous one. If both situations $r <$0.5 but $|E| \geq$0.5 occur, the rabbit is fully prepared for a successful escape because its energy is sufficient. Before a surprise and rapid pounce, the Harris hawk makes a soft besiege.
   The Levy Flight (LF) mimics the zigzag escape patterns of rabbits.
   A soft siege is performed according to the following rule, as outlined in Eq. (2.14), can decide what to do:

$$Y = X_{rabbit}(t) - E\,|JX_{rabbit}(t) - X(t)|\,. \tag{2.14}$$

   If the prey's movements become more deceptive, the hawks will dive abruptly and irregularly. These dives are based on levy flight patterns, as outlined in the subsequent rule:

$$Z = Y + S \times LF(D). \tag{2.15}$$

   S is a random vector whose size is $1 \times D$ D indicates the problem's dimension. LF is determined by applying Eq. (2.16) [8]:

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1+\beta) \times sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}} \right)^{\frac{1}{\beta}}. \tag{2.16}$$

β is set to 1.5 which is a predetermined constant, u, v are random values that are selected in (0,1). Then, in the soft besiege phase, the hawks' position is updated by Eq. (2.17):

$$X(t+1) = \begin{cases} Y & if F(Y) < F(X(t)) \\ Z & if F(Z) < F(X(t)). \end{cases} \tag{2.17}$$

By using Eq.(2.14) and (2.15), Z and Y are obtained [8].

(d) Hard besiege combined with increasing surprise and fast dives:

When $|E|$ <0.5 and r <0.5, the rabbit is unable to escape because it dœs not have enough energy. Then before the surprise pounce, a hard besiege is established to catch the rabbit successfully. To model it mathematically in hard besiege condition, the following rule is performed :

$$X(t+1) = \begin{cases} Y & if F(Y) < F(X(t)) \\ Z & if F(Z) < F(X(t)). \end{cases} \tag{2.18}$$

using the updated formula (Equation (2.19) and (2.20)), Z and Y are obtained:

$$Y = X_{rabbit}(t) - E\,|JX_{rabbit}(t) - X_m(t)| \tag{2.19}$$

$$Z = Y + S \times LF(D). \tag{2.20}$$

Equation (2.9) yields $X_m(t)$.

The HHO algorithm is shown via Algorithm 2.

Algorithm 2 HHO algorithm [8]

Inputs: Maximum number of iterations (T), the size of the population (N)
Outputs: Rabbit's location, corresponding fitness

1. Randomly initialize the population $X_i(i = 1, 2, \ldots, N)$
2. **While**(termination condition is not satisfied) **do**
   (a) Evaluate the fitness of each hawk in the population
   (b) Assign the rabbit's optimum position (best location) to X_{rabbit}
   (c) **for** (each hawk (X_i)) **do**
      i. Update the jump strength J and beginning energy E_0.
      ii. Use Eq. (2.10) to update the E
      iii. **if** ($|E| \geq 1$) **then**
      Update the vector of locations based on Eq. (2.8)
      end if
      iv. **if** ($|E| < 1$) **then**

      if ($r \geq$0.5 and $|E| \geq 0.5$) **then**
      A. Use Eq. (2.11)to update the location vector
      B. **else if** ($r \geq$0.5 and $|E| < 0.5$) **then**
      Use Eq. (2.13) to update the location vector
      C. **else if** ($r <$0.5 and $|E| \geq 0.5$) **then**
      Use Eq. (2.17) to update the location vector
      D. **else if** ($r <$0.5 and $|E| < 0.5$) **then**
      Use Eq. (2.18)to update the location vector
      end if
      end if
      end for
   end while
3. **Return** X_{rabbit}

3 The proposed method

This section will offer a thorough explanation of the proposed method that uses meta-heuristic methods. There are numerous meta-heuristic methods accessible nowadays. The basic iterations of nature-inspired algorithms that are currently in use have advanced recently. As a result of these modifications, they can be applied to a variety of challenging optimization problems. They make use of random variables and a variety of parameters that need to be carefully defined in order to solve an issue effectively. Each approach offers distinct benefits in terms of

flexibility, performance under uncertainty, and unexplored search areas [18]. The proposed method is derived from SMA and HHO based clustering. Both algorithms have shown promising results in clustering tasks by optimizing the placement and distribution of data points within different clusters. In the SMA based clustering algorithm, the centers of the clusters are considered as the positions of the slime moulds. In fact, when the algorithm terminates and the slime mould reaches the food source, the algorithm outputs the optimal centers. This method is able to find suitable centers, leading to accurate and efficient clustering results. Also, in the HHO based clustering algorithm, the Harris hawk's attempt to reach the prey is actually an attempt to reach the best centers for clustering. So optimal centers have been obtained when the prey is hunted. The HHO algorithm employs the hunting behavior and social interaction of birds to optimize the clustering process. Through simulations and mathematical models, this algorithm exhibits significant improvements in terms of convergence speed and solution quality over existing metaheuristic algorithms, which identifies clusters, resulting in improved clustering speed.

Even while SMA and HHO work well in a variety of situations, they both need constant enhancements to become more resilient and flexible. Among the challenges facing the SMA and HHO algorithms are the possibility of getting stuck in local optima and the need for additional processing power in high-dimensional issues. Also achieving the best results requires fine-tuning the balance between exploration and exploitation, because the requirement for a wider search space coverage can impede performance. Combining various algorithms could maximize their benefits and reduce their drawbacks, which would ultimately result in quicker and more precise results. In this method, a new hybrid methodology is employed to locate the best cluster centers, which incorporates computational intelligence techniques to solve this problem. In order to improve search space exploration and exploitation, it also integrates hybrid mechanisms that draw inspiration from SMA and HHO behavior. The combination of different approaches could lead to enhanced accuracy, and increased robustness and better convergence as compared to the separate algorithms in solving complex problems during the algorithm. The algorithm starts and the centers of each cluster are initialized. SMA algorithm will be called to facilitate the initial groundwork. Following the completion of this stage, the optimal solution for the entire problem is compared with the best result based on the objective function value. The optimal solution is updated in accordance with any improvements that are noticed. The best outcomes from the SMA algorithm are then chosen and added to the HHO algorithm's input. In HHO, these chosen outcomes take the role of the initial random population. The best SMA results are then positioned at the top of the initial population list in HHO rather than creating a random starting population. After that, randomly generated solutions are used to fill in the remaining spaces. This modified initial population is used to execute HHO. The HHO algorithm will then is called to harmoniously complement the previous outputs. This will create an exquisite procedural flow that encompasses a synergy of sequential hermeneutics. Then the fitness values are compared. If an improvement is noticed, the optimal solution is updated once more. We choose the best option identified by HHO from the top results. These variables are added back into the SMA algorithm, and the identical method used for HHO is used to create the initial population for SMA. So the best solution is transferred to the next step and they are entered into the algorithms as primary solution. This process terminates when the algorithm's termination condition is met. In summary, the output of each step is used as the input for the subsequent step, and the previous method is carried over at the start of each iteration. By analyzing the outcomes and taking the objective function value into account, the algorithm finds the cluster centers that are most suited for the data. Figure 1 depicts the process for enhancing clustering, while Algorithm 3 displays the proposed algorithm.

Talbi [34] introduced multiple techniques for hybridizing heuristic algorithms. As mentioned in [34], it is possible to hybridize two algorithms using high-level or low-level integration, employing relay or cœvolutionary methods in either homogeneous or heterogeneous approaches. In this article, two algorithms are hybridized using HRH (High-level relay hybrid) and heterogeneous hybrid. The hybrid is HRH because utilizes a sequential execution of self-contained metaheuristics. It is really because they are implemented consecutively, where each subsequent metaheuristic takes the output of the preceding one

as its input, operating in a sequential pipeline fashion. It is heterogeneous because this method arises from the utilization of two distinct algorithms to ultimately generate the desired outcomes.
The best cluster centers can be found using optimization algorithms, which are designed to find the best solution or the best possible values for the specific problem parameters. Optimization aims to reduce or increase an objective function while satisfying certain constraints; in this algorithm, the objective function is determined by reducing the intra-cluster distance. Finding the most appropriate cluster centers is crucial for improving clustering, because it reduces the distance within clusters. In the proposed method the following are considered:

1. For each O_i data, calculate the Euclidean distance of that data to each C_j center of the cluster.

2. In each step, assign the data O_i to the closest cluster S_k, $k = \{1, 2, ..., K\}$.

3. In this paper, the objective is minimization.

4. Calculate the fitness value as follows:

$$F = \sum_{i=1}^{N} \sum_{j=1}^{K} W_{ij} \|O_i - C_j\|^2 \tag{3.1}$$

W is set in a special way in each of the SMA and HHO. The hybrid HHO-SMA algorithm's procedure is demonstrated in Figure 1.

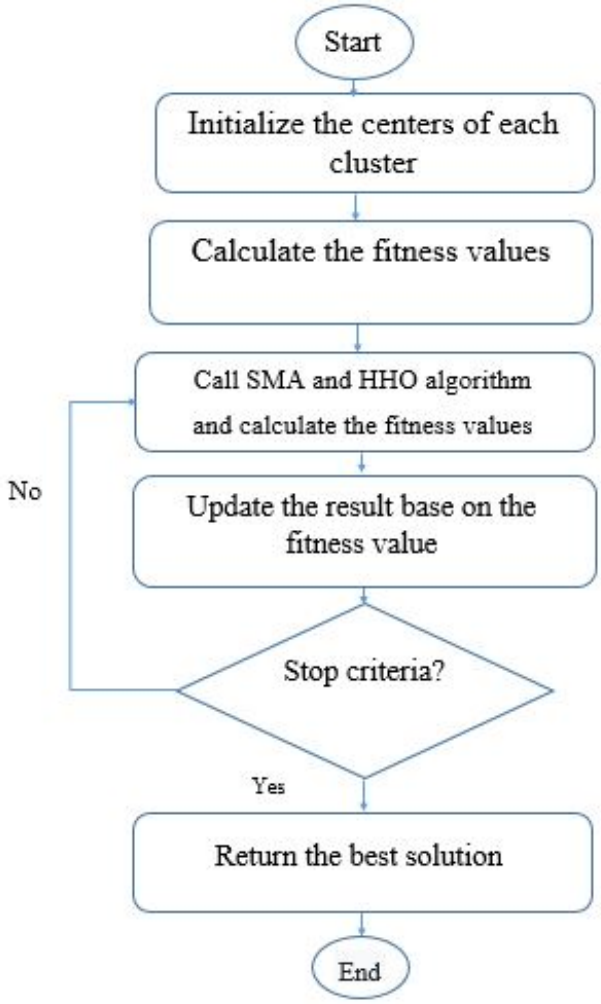


Figure 1: Steps of hybrid HHO-SMA algorithm.

The proposed method is presented in Algorithm 3. Here tol is described as the distinction between the fitness function's value in the current iteration and the previous iteration.

Algorithm 3 Proposed hybrid HHO-SMA algorithm

Inputs: Number of clusters (K), population size (N), maximum tolerance (ε), data dimensions (D), the maximum number of iterations (max_it)
Outputs: The best fitness value and optimal centers

1. Initialize all center $C_i(i = 1, 2, \ldots, K)$ randomly
2. Evaluate the fitness value using Eq. (3.1)
3. $T = 0$
4. **While**($tol \geq \ \varepsilon \,\&\&\, T \leq \ max\_it$) **do**
   (a) Call the HHO and SMA algorithms
   (b) Calculate the fitness values using Eq. (3.1)
   (c) Update the centers base on the fitness values
   (d) Calculate tol
   (e) $T = T + 1$
5. **Return** The optimal centers and best fitness value

HHO's advantages in exploring and making use of search spaces are successfully combined by the HHO-SMA algorithm with extra tactics motivated by the SMA's biological behaviors. The goal of this hybrid strategy is to improve optimization performance, especially for complicated problems. By combining the best solutions identified in previous stages with newly generated solutions at each iteration, a balance is achieved between exploring novel areas of the search space and exploiting promising regions discovered in prior iterations, ensuring both diversity and convergence in the solution search. In fact, this mechanism improves the exploration of the search space and reduces the risk of premature convergence.

4 Experimental results

All algorithms were executed on a Windows 10, 64-bit professional computer equipped with 16 GB of RAM, using MATLAB R2016b. To evaluate the proposed method, two sets of experiments were conducted, which will be explained in detail later.

4.1 Benchmark set and comparison algorithms

From the optimization literature, a number of popular benchmark functions are chosen and utilized to examine the HHO-SMA algorithm's numerical efficiency [37, 4, 36]. Each benchmark function may highlight different aspects of the HHO-SMA algorithm's capabilities. Functions F1 to F7 assess the algorithm's exploitation capabilities (intensification) with a single global optimum, while standard functions F8 to F23 evaluate the algorithm's exploration capabilities (diversification). The Tables 4,5,6 sum up the test problems and attributes of the functions. The cost function, optimization variable range of variation, and

optimal value f_{min} as defined in the literature are reported. The HHO-SMA, HHO and SMA algorithms utilized a population of 30 individuals and a maximum of 500 iterations. The rest of the parameters were set as mentioned in subsections 2.1 and 2.2 and summarized in Tables **??**, 2, and 3. The proposed algorithm maintains this parameter configuration, referencing the provided these Tables as needed. Random factors affect the results in the algorithm. To solve this problem, For each function, each method was run ten times separately, and the average of those runs was used to determine the outcome. Worst, Best, and AVG (average findings) were used to measure the experiment's outcomes. Results of comparisons on 23 benchmark functions are shown in Table 7, including scalable F1-F23 functions and using 30 Dimensions. The proposed method is evaluated on three datasets: Iris, Wine, and Vowel. For the majority of functions (e.g., F1, F2, F3, F6, F7, etc.), the best values are obtained using the proposed HHO-SMA algorithm. As an example, in F1, the best value obtained by HHO-SMA is 1.6217E-43, which is substantially better than 9.2725E-16 for HHO and 8.7163E-10 for SMA. Even under the most direct circumstances, HHO-SMA performs dependably. The worst value for HHO-SMA in F6, for instance, is 2.4935E-05, whereas HHO and SMA produce significantly worse outcomes at 5.2072E-02 and 7.4881E-02, respectively. HHO-SMA routinely performs better than the other approaches in terms of average values. The average value for HHO-SMA in F7, for example, is 2.1520E-05, whereas the average values for HHO and SMA are 3.2455E-03 and 3.8790E-01, respectively. The HHO-SMA algorithm continues to demonstrate its advantages for complex tasks such as F10 and F12. In particular, the best value found in F10 is 8.8818E-16, which is almost zero and indicates an ideal solution. In comparison to HHO and SMA, HHO-SMA exhibits higher stability in reaching the best, worst, and average values across all functions. This demonstrates its resilience and dependability in resolving a variety of optimization issues. When compared to the other approaches, the HHO-SMA algorithm constantly avoids becoming stuck in local optima and produces better results for complicated and nonlinear functions.
This investigation shows that HHO-SMA performs better than other methods in every criterion, making it a very dependable and successful method for handling challenging optimization issues. In scholarly articles, this benefit could be highlighted as a noteworthy contribution.

4.2 Analysis of qualitative results of the HHO-SMA algorithm on unimodal and multimodal functions

Figure 2 shows the qualitative outcomes of HHO-SMA for a number of conventional unimodal and multimodal cases on four distinct issues (F2, F3, F12, and F15), each of which represents an objective function with a different degree of complexity. A convergence curve and a 3D parameter space chart are included in each set to illustrate the algorithm's convergence speed and qualitative outcomes.

4.2.1 Chart interpretation

1. F2 and F3 problems (complex and multi-modal functions)

    (a) Parameter space chart: Multiple peaks and troughs are seen in these functions, suggesting significant complexity and a large number of local minima. The proposed method successfully traverses the search space and approaches the global minimum in spite of this complexity.

    (b) Convergence curve: The algorithm rapidly lowers the objective function value, as evidenced by the sharp drop in error in the initial iterations. The error successfully converges to the

Table 1: Key parameters of HHO algorithm and their recommended values

Parameter	Description	Recommended Value
Population size (N)	Number of individuals in the initial population	$10 \leq N \leq 100$ (commonly 30)
Number of iterations (T)	Total number of iterations	$100 \leq T \leq 1000$ (commonly 500)
Learning rate (r)	A random value for selecting flight behaviors	Randomly generated in $[0, 1]$
Distance to prey ($X_{\text{prey}} - X_{\text{hawk}}$)	Difference between the positions of hawks and the prey	Dynamic, no fixed value
Prey energy factor (E)	Linearly decreases from a positive to a negative value	$E = 2(1 - \text{Iteration}/\text{Max Iterations})$
Levy flight	Creates diversity and randomness in search	Calculated using Levy distribution
Attack and escape rate (p)	Probability of selecting between attack and chase strategies	Randomly generated in $[0, 1]$

Table 2: Key parameters of SMA algorithm and their recommended values.

Parameter	Description	Recommended Value
Population size (N)	Number of individuals in the initial population	$10 \leq N \leq 100$ (commonly 30)
Number of iterations (T)	Total number of iterations	$100 \leq T \leq 1000$ (commonly 500 or 1000)
Weight factor (W)	Influences the behavior of slime mould	$W = \frac{1}{\lvert \text{fitness}_i - \text{fitness}_{\text{best}} \rvert}$
Control parameter (a)	Linearly decreases during iterations	$a = 1 - \frac{\text{Iteration}}{\text{Max Iterations}}$
Random chance (r)	Random value for exploration or exploitation	Randomly generated in $[0, 1]$
Control parameter (p)	Probability for decision-making	$p = \frac{\tanh \lvert \text{fitness}_i - \text{fitness}_{\text{best}} \rvert}{\text{Iteration}}$
Search direction	Determines movement strategy	Dynamic based on $r < p$ or $r \geq p$

Table 3: Key parameters of HHO-SMA algorithm and their recommended values.

Parameter	Description	Recommended Value
Population size (N)	Number of individuals in the initial population	$10 \leq N \leq 100$ (commonly 30)
Number of iterations (T)	Total number of iterations	$100 \leq T \leq 1000$ (commonly 500)
Learning rate (r)	A random value for selecting flight behaviors	Randomly generated in $[0, 1]$
Distance to prey ($X_{\text{prey}} - X_{\text{hawk}}$)	Difference between the positions of hawks and the prey	Dynamic, no fixed value
Prey energy factor (E)	Linearly decreases from a positive to a negative value	$E = 2(1 - \text{Iteration}/\text{Max Iterations})$
Levy flight	Creates diversity and randomness in search	Calculated using Levy distribution
Attack and escape rate	Probability of selecting between attack and chase strategies	Randomly generated in $[0, 1]$
Weight factor (W)	Influences the behavior of slime mould	$W = \frac{1}{\lvert\text{fitness}_i - \text{fitness}_{\text{best}}\rvert}$
Control parameter (a)	Linearly decreases during iterations	$a = 1 - \frac{\text{Iteration}}{\text{Max Iterations}}$
Random chance	Random value for exploration or exploitation	Randomly generated in $[0, 1]$
Control parameter (p)	Probability for decision-making	$p = \frac{\tanh\lvert\text{fitness}_i - \text{fitness}_{\text{best}}\rvert}{\text{Iteration}}$
Search direction	Determines movement strategy	Dynamic based on $r < p$ or $r \geq p$

Table 4: Information on unimodal benchmark functions.

No	Function	Dim	Range	Fmin
F1	$f(x) = \sum_{i=1}^{d} x_i^2$	30	$[-100, 100]^d$	0
F2	$f(x) = \sum_{i=1}^{d} \lvert x_i \rvert + \prod_{i=1}^{d} \lvert x_i \rvert$	30	$[-10, 10]^d$	0
F3	$f(x) = \sum_{i=1}^{d} (\sum_{j=1}^{i} x_j)^2$	30	$[-100, 100]^d$	0
F4	$f(x) = \max_i \{\lvert x_i \rvert\}, 1 \leq i \leq d$	30	$[-100, 100]^d$	0
F5	$f(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]^d$	0
F6	$f(x) = \sum_{i=1}^{d} (\lvert x_i + 0.5 \rvert)^2$	30	$[-100, 100]^d$	0
F7	$f(x) = \sum_{i=1}^{d} i x_{4i} + \text{random}[0, 1)$	30	$[-128, 128]^d$	0

ideal solution after a few iterations when it stabilizes at a constant value. These outcomes demonstrate how well the approach handles the difficulties presented by multi-modal functions.

2. F12 and F15 problems (simpler and unimodal functions)

   (a) Parameter space chart: These functions are simpler to optimize since they show distinct global minima without notable local minima. These functions' structures show that the algorithm takes a more direct and effective route to the ideal location.

   (b) Convergence curve: The algorithm's exceptional efficiency in finding the best solution is demonstrated by the steep slope and quick error reduction during early iterations and the convergence curve depicts the best fitness value obtained in HHO-SMA during the iteration process. They demonstrate that HHO-SMA exhibits strong convergence capabilities with a fast decrease in cost and the convergence curves exhibit a clear downward trajectory, indicating fast convergence. The method's dependability for easier problems is demonstrated by the speed and exceptional stability of the convergence.

4.2.2 Advantages of the proposed method based on the results:

1. High performance on complex functions: The HHO-SMA algorithm converges to the global minimum and effectively manages local minima in complex problems (such as F2 and F3). This demonstrates how well the approach can handle multi-modal optimization challenges.

2. Rapid convergence speed: The algorithm's quick learning rate and versatility are demonstrated by the notable decrease in error across all problems, both basic and complicated.

3. Stability in achieving the global optimum: The proposed method converges more quickly in smaller situations (like F12 and F15) and finds exact optima.

Table 5: Descriptions of the multimodal benchmark functions.

No	Function	Dim.	Range	Fmin
F8	$f(x) = \sum_{i=1}^{d} \left(x_i \sin\left(\sqrt{\lvert x_i \rvert} \right) \right)$	30	[-500, 500]d	$-418.9829 \times n$
F9	$f(x) = 10d + \sum_{i=1}^{d} \left[x_i^d - 10\cos(2\pi x_i) \right]$	30	[-5.12, 5.12]d	0
F10	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2} \right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d} \cos(2\pi x_i) \right) + 20 + e$	30	[-32, 32]d	0
F11	$f(x) = \frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	30	[-600, 600]d	0
F12	$f(x) = \frac{\pi}{d}\left\{ 10\sin(\pi y_1) + \sum_{i=1}^{d-1} (y_i - 1)^2 \left[1 + 10\sin^2(\pi y_{i+1}) \right] + (y_d - 1)^2 \sum_{i=1}^{d} U(x_i, 10, 100, 4) \right\}$	30	[-50, 50]d	0
F13	$f(x) = 0.1\left\{ \sin^2(3\pi x_1) + \sum_{i=1}^{d} (x_i - 1)^2 \left[1 + \sin^2(3\pi x_i + 1) \right] + (x_d - 1)^2 \right\} + \sum_{i=1}^{d} U(x_i, 5, 100, 4)$	30	[-50, 50]d	0

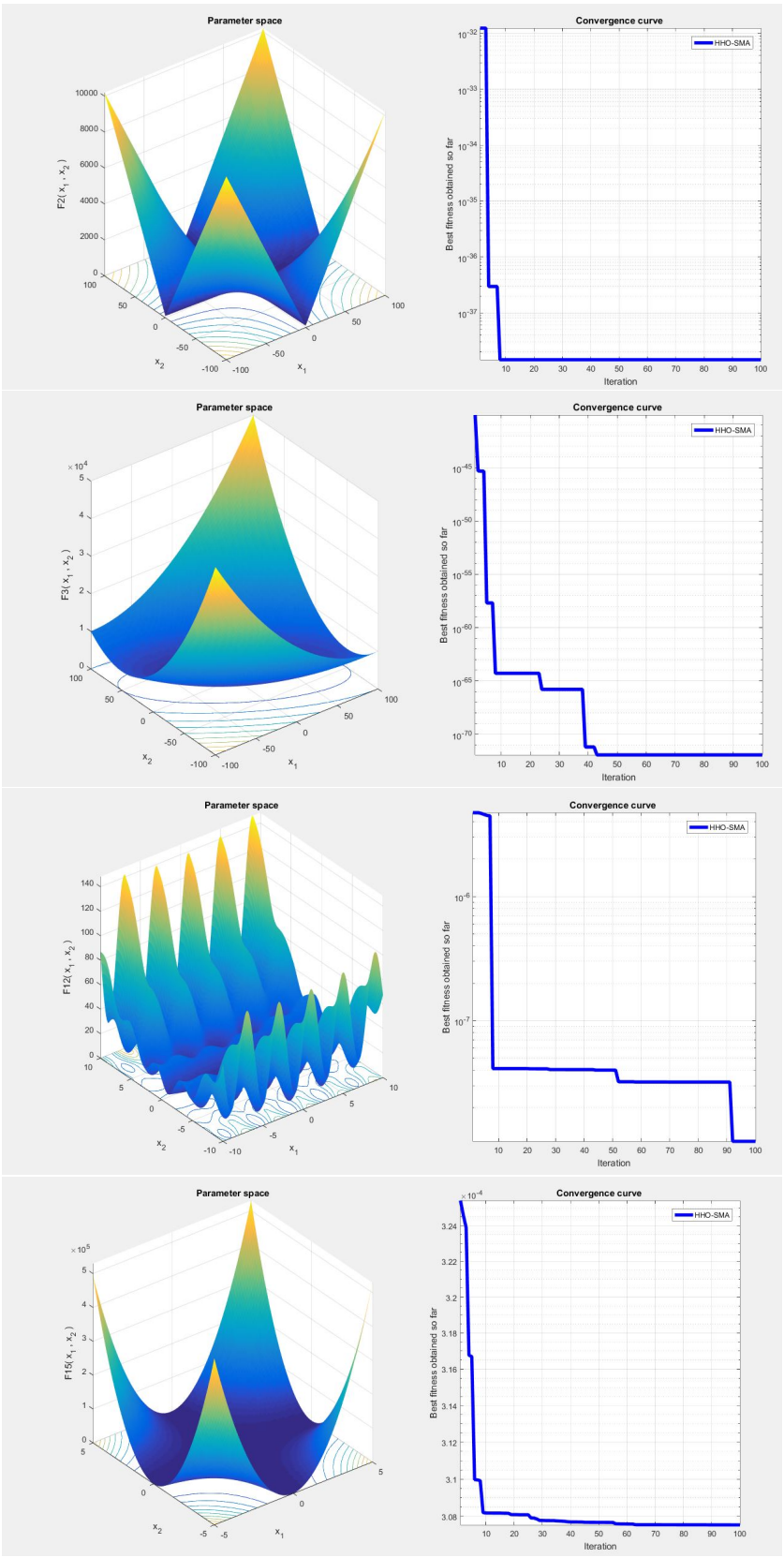


Figure 2: Qualitative results for F2, F3, F12, F15 problem.

Table 6: Description of multimodal benchmark functions with fixed dimensions.

No	Function	Dim.	Range	Fmin
F14	$f(x) = \left[\frac{1}{500} + \sum_{i=1}^{25} \frac{1}{i+\sum_{j=1}^{2}(x_j - a_{j,i})^6}\right]^{-1}$	30	$[-65, 65]^d$	-1
F15	$f(x) = \sum_{i=1}^{d} [a_i - \frac{x_1(b_{2i}+b_i x_2)}{b_i{}^2 + b_i x_3 + x_4}]^2$	30	$[-5, 5]^d$	0.00030
F16	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	30	$[-5, 5]^d$	-1.0316
F17	$f(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	30	$[-5, 5]^d$	0.398
F18	$f(x) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right]$ $\times \quad \left[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right]$	30	$[-2, 2]^d$	3
F19	$f(x) = -\sum_{i=1}^{4} a_i \exp\left(-\sum_{j=1}^{3} b_{ij}(x_j - p_{ij})^2\right)$	30	$[1, 3]^d$	-3.86
F20	$f(x) = -\sum_{i=1}^{4} a_i \exp\left(-\sum_{j=1}^{6} b_{ij}(x_j - p_{ij})^2\right)$	30	$[0, 1]^d$	-3.32
F21	$f(x) = -\sum_{i=1}^{5} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	30	$[0, 10]^d$	-10.1532
F22	$f(x) = -\sum_{i=1}^{7} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	30	$[0, 10]^d$	-10.4028
F23	$f(x) = -\sum_{i=1}^{10} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	30	$[0, 10]^d$	-10.5363

4.3 Evaluation of algorithm performance in clustering problem

In the second test, analyzes the numerical results obtained from the experiments to demonstrate the effectiveness of the proposed HHO-SMA method in enhancing clustering performance compared to other optimization algorithms. all the experiments have been done on tree data sets used for the clustering problem: Iris flowers, Vowel and Wine. Iris flowers contain four features for three species of Iris flowers, Vowel contains tree features and Wine contains 13 attributes.

In these experiments, slime mould and Harris hawks algorithms were used for clustering. Several similarity metrics can be used for solution clustering. Euclidean distance was selected for its efficiency in this research. The average of fitness values of these three algorithms has been evaluated in 10 iterations to find the optimal centers.

4.3.1 Comparative the numerical analysis of the proposed method

Table 8 illustrates the proposed algorithm's efficacy and presents the fitness values obtained by the HHO-SMA algorithm against other optimization algorithms such as HHO, SMA, GA, SSA, and ABC. The evaluation considers the mean, best, and worst fitness values across multiple independent runs, reflecting both the quality and robustness of the algorithms under varying conditions.

By obtaining the lowest mean fitness values as compared to alternative techniques, the HHO-SMA algorithm typically performs better. This demonstrates its improved capacity to find more precise and ideal data clustering solutions. The algorithm's stability and resilience are demonstrated by HHO-SMA's constantly lower mean fitness values throughout various runs. For dependable and consistent clustering results in real-world applications outside of controlled laboratory conditions, this stability is especially crucial.

Table 7: Performance evaluation of benchmark functions (F1-F23) with 30 dimensions.

No.Function	Type	HHO-SMA	HHO	SMA
F1	Best	1.6217E-43	9.2725E-16	8.7163E-10
	Worst	7.6731E-33	1.3418E-13	1.7240E-01
	Ave	9.0978E-34	2.1329E-14	1.9817E-02
F2	Best	1.0899E-20	1.0304E-11	5.4268E-15
	Worst	7.0811E-18	2.4794E-07	1.1988E-11
	Ave	2.6767E-18	7.04392E-08	1.9515E-12
F3	Best	3.9520E-35	5.2278E-16	3.3257E-09
	Worst	2.1378E-31	3.3052E-10	5.5622E+00
	Ave	2.3379E-32	5.5664E-02	5.5664E-01
F4	Best	4.0156E-19	1.4000E-09	3.8296E-13
	Worst	4.2005E-17	5.8974E-07	3.0585E-09
	Ave	9.9820E-18	1.3311E-07	3.1157E-10
F5	Best	2.8824E-10	6.0434E-03	2.8908E+01
	Worst	3.5765E-03	7.4237E+00	2.9007E+01
	Ave	1.4445E-03	2.2802E+00	2.8980E+01
F6	Best	2.8541E-07	2.9884E-04	6.1202E-04
	Worst	2.4935E-05	5.2072E-02	7.4881E-02
	Ave	7.3742E-06	1.9496E-02	6.9790E-02
F7	Best	2.3859E-06	3.0007E-04	1.1313E-02
	Worst	3.6033E-05	7.5128E-03	9.5780E-02
	Ave	2.1520E-05	3.2455E-03	3.8790E-01
F8	Best	-1.2569E+04	-1.2556E+04	-1.2534E+04
	Worst	-1.2569E+04	-8.9961E+03	-2.3848E+04
	Ave	-1.2569E+04	-1.9431E+04	-6.8852E+04
F9	Best	0	0	0
	Worst	2.7346E-09	2.9803E-01	2.1607E-01
	Ave	4.8981E-10	3.5187E-03	3.2705E-3
F10	Best	8.8818E-16	4.0735E-09	1.0040E-09
	Worst	8.8818E-16	1.3030E-06	1.3797E-05
	Ave	8.8818E-16	2.2947E-07	4.1836E-06
F11	Best	0	0	0
	Worst	5.3180E-14	3.6393E-12	3.3734E-10
	Ave	6.1069E-15	5.9932E-13	3.3739E-11
F12	Best	1.0948E-07	1.7194E-05	1.0113E+00
	Worst	3.9953E-06	3.0448E-03	1.6112E+00
	Ave	2.0851E-06	7.9794E-04	1.3102E+00
F13	Best	6.2543E-16	1.0141E-04	2.9970E+00
	Worst	7.7229E-05	1.0992E-02	2.9996E+00
	Ave	2.3275E-05	3.8392E-03	2.9988E+00
F14	Best	9.9800E-01	9.9800E-01	9.9980E-01
	Worst	9.9800E-01	5.9289E+00	1.2670E+01
	Ave	9.9800E-01	2.3813E+00	9.8623E+00
F15	Best	3.0751E-04	3.2785E-04	7.0057E-04
	Worst	3.0923E-04	3.6630E-03	1.4228E-01
	Ave	3.0802E-04	1.2039E-03	3.7124E-02
F16	Best	-1.0316E+00	-1.0316E+00	-1.0316E+00
	Worst	-1.0316E+00	-1.0316E+00	-1.0316E+00
	Ave	-1.0316E+00	-1.0316E+00	-1.0316E+00
F17	Best	3.9789E-01	3.9792E-01	4.0398E-01
	Worst	3.9789E-01	4.0593E-01	4.6879E+00
	Ave	3.9789E-01	3.6083E-01	9.74801E-01
F18	Best	3.0000E+00	3.0000E+00	3.0000E+00
	Worst	3.0000E+00	3.0000E+00	3.0000E+00
	Ave	3.0000E+00	3.0000E+00	3.0000E+00
F19	Best	-3.8628E+00	-3.8291E+00	- 3.8564E+00
	Worst	-3.8628E+00	-3.5560E+00	-8.7176E-01
	Ave	-3.8628E+00	-3.7620E+00	- 3.3346E+00
F20	Best	-3.3220E+00	-2.9838E+00	-2.6266E+00
	Worst	-3.2031E+00	-2.2571E+00	-2.8649E-01
	Ave	-3.2625E+00	-2.6660E+00	-4.017944E-01
F21	Best	-10.1532E+00	-5.0507E+00	-5.4992E+00
	Worst	-10.1532E+00	-4.8468E+00	-6.9580E-01
	Ave	-10.1532E+00	-4.9806E+00	-2.0076E+00
F22	Best	-10.4029E+00	-5.0371E+00	- 5.0855E+00
	Worst	-10.4029E+00	-2.4242E+00	- 6.9917E-01
	Ave	-10.4029E+00	-4.6530E+00	- 2.9785E+00
F23	Best	-10.5364E+00	-5.1201E+00	-1.0027E+01
	Worst	-10.5364E+00	-4.9134E+00	- 5.9359E-01
	Ave	-10.5364E+00	-4.9432E+00	- 3.9718E+00

The best fitness value (33.2476) and average fitness value (35.6547) on the Iris dataset show how well the HHO-SMA approach performs when compared to other algorithms and how accurate and efficient it is, even on relatively basic datasets.

On the Vowel dataset with 5 clusters, the HHO-SMA approach performs marginally better than the other methods, with a best fitness value of 1.5808E+05. Additionally, HHO-SMA's average fitness is 1.7810E+05, which is superior to other models. This illustrates how the HHO-SMA method can effectively cluster data in situations when the number of clusters is more in line with the data structure and extract the inherent data structure. The average fitness value of the HHO-SMA technique on the Vowel dataset with 6 clusters is 1.7870E+05, which is substantially better than the other approach, such as the GA method's comparable value of 1.9961E+05. Because of this enhancement, the HHO-SMA approach typically produces better and more reliable results. In comparison to the GA method's best fitness value of 1.4517E+05, the HHO-SMA method's best fitness value of 1.5716E+05 is marginally lower. Though both approaches can produce desired results, the HHO-SMA method provides higher overall stability, as evidenced by the little difference between the two methods' best fitness values. The HHO-SMA method's difference between the best and worst fitness values is 5.841E+04, whereas the GA method's difference is 9.591E+04. This suggests that the HHO-SMA approach performs more constantly and with less variation than GA across various runs. The equivalent number in GA (2.4911E+05) is substantially worse than the worst fitness value in the HHO-SMA technique (2.1636E+05). This benefit shows that even in the worst-case situations, the HHO-SMA method performs satisfactorily, whereas the GA method has occasionally produced inferior outcomes.

Likewise, this approach yields the highest fitness value (1.0569E+04) in the Wine dataset, demonstrating its strong data clustering capabilities. The HHO-SMA technique generally maintains competitive and frequently better performance on more complex datasets like Wine and Vowel, which are characterized by higher dimensionality and hence increased clustering complexity. This demonstrates the HHO-SMA algorithm's resilience, flexibility, and adaptation while working with datasets of different degrees of complexity.

By cleverly integrating the advantages of the HHO and SMA algorithms, HHO-SMA delivers notable gains, even though the standalone methods also produce results that are passably decent. HHO-SMA can more successfully traverse the search space and stay out of local optima because of this synergistic combination. Among all datasets, the HHO-SMA approach shows the lowest worst fitness value. For instance, a score of 38.6942 in the Iris dataset and a value of 1.8881E+05 in the Vowel dataset show that this strategy is effective even in adverse circumstances. The HHO-SMA method's strong stability across a range of datasets is demonstrated by the tiny difference between its best and worst fitness values. When compared to HHO-SMA, the other comparison algorithms SSA, ABC, DE, and GA generally perform worse, suggesting that they are less able to efficiently explore the solution space in certain clustering circumstances.

The number of clusters affects the performance of every algorithm. As demonstrated in the Wine and Vowel datasets, the HHO-SMA algorithm maintains its strong and consistent performance as the number of clusters increases (for example, from three to five or six clusters). This demonstrates even more how flexible and appropriate this approach is for more complex clustering problems.

In conclusion, the results show that, when compared to other methods, the HHO-SMA algorithm constantly produces the best (lowest) fitness values, the least amount of variation in outcomes between runs, and the highest degree of stability. The HHO-SMA method successfully avoids premature convergence to local optima and demonstrates improved exploration capabilities. These results demonstrate that HHO-SMA is a strong, effective, and trustworthy method for clustering tasks in a variety of datasets

and environments. The following is a summary of the benefits of the proposed method based on the findings of these experiments:

1. Effective search space exploration: The HHO-SMA algorithm's combination of HHO and SMA results in a more efficient search space exploration process. This enables the algorithm to bypass local optima where other algorithms could become trapped and to seek for optimal solutions more thoroughly.

2. Stability and adaptability: This algorithm's high reliability is demonstrated by the high stability of HHO-SMA findings across many runs. It is a useful tool for a variety of clustering problems due to its adaptability to different datasets with varying levels of complexity.

3. Improvement over base algorithms: HHO-SMA demonstrates significantly better performance compared to the base HHO and SMA algorithms. This demonstrates the success of combining these two algorithms and creating a more powerful optimization method.

4. Superior performance on complex datasets: HHO-SMA has remarkable performance, particularly on complex datasets with high dimensionality and more clustering difficulties, such as Wine and Vowel. This illustrates the algorithm's exceptional capacity to tackle difficult issues.

4.3.2 Superiority of the proposed HHO-SMA algorithm in Silhouette score-based data clustering

The results of comparing the Silhouette score across several standard clustering algorithms and the proposed hybrid HHO-SMA method are shown in Table 9. This comparison is performed on four different datasets: Iris, Vowel, Wine, and Wisconsin Diagnostic Breast Cancer (WDBC). The compared algorithms include k-means, hierarchical clustering (Ward's method), DBSCAN, spectral clustering, the Gaussian Mixture Model (GMM), and the proposed HHO-SMA method. The Silhouette score is a statistic used to assess the quality of clustering results, based on two main concepts: intra-cluster cohesion and inter-cluster separation. Essentially, it measures how closely data points resemble their designated cluster and how different they are from other clusters. The range of the Silhouette score is -1 to +1. Data points are closely clustered inside their respective groups and well-separated from other clusters when the score is near +1, which denotes a well-defined clustering structure. A score close to 0 indicates unclear cluster assignment since it indicates that the data points are situated close to the decision border between clusters. A score near -1, on the other hand, indicates improper clustering and suggests that data points are probably assigned to the incorrect cluster [29].
With scores of 0.68 and 0.67, respectively, traditional techniques like k-means and hierarchical clustering performed reasonably well on the Iris dataset which is comparatively simple dataset. At a score of 0.70, spectral clustering fared better. With a score of 0.85, the proposed HHO-SMA approach, however, showed a notable improvement above the best standard algorithm, representing a performance gain of about 25%.
The algorithms had a greater challenge on the Vowel dataset because it is more complicated. With scores of 0.42 and 0.44, respectively, conventional techniques like k-means and hierarchical clustering showed a sharp fall in performance. With a score of 0.46, spectral clustering outperformed the other common approaches; however, HHO-SMA outperformed it by 13%, with a score of 0.52.
Traditional techniques like GMM and k-means obtained scores of 0.55 and 0.57 on the Wine dataset,

Table 8: Comparison of fitness values performance.

Algorithm	Data	Ave. of fitn. value	The best fitn. value	The worst fitn. value	No. iter	No. clusters
ABC	Iris	$3.5667E+02$	$1.1585E+02$	$3.8343E+02$	10	3
SSA	Iris	$1.8314E+02$	$1.0274E+02$	$2.2965E+02$	10	3
GA	Iris	$1.1541E+02$	$9.8259E+01$	$1.3475E+02$	10	3
SMA	Iris	$4.5121E+01$	$4.2010E+01$	$6.3385E+01$	10	3
HHO	Iris	$7.7144E+01$	$6.4348E+01$	$1.0151E+02$	10	3
$HHO-SMA$	Iris	$3.8123E+01$	$3.5768E+01$	$4.0305E+01$	10	3
ABC	Iris	$3.9299E+02$	$2.2988E+02$	$4.1111E+02$	10	5
SSA	Iris	$3.4920E+02$	$2.2097E+02$	$3.8181E+02$	10	5
GA	Iris	$2.0539E+02$	$1.5391E+02$	$2.8176E+02$	10	5
SMA	Iris	$4.3250E+01$	$4.0891E+01$	$5.5706E+01$	10	5
HHO	Iris	$7.5014E+01$	$6.2398E+01$	$8.4743E+01$	10	5
$HHO-SMA$	Iris	$3.5654E+01$	$3.3247E+01$	$3.8694E+01$	10	5
ABC	Vowel	$2.6668E+05$	$2.0401E+05$	$2.7364E+05$	10	5
SSA	Vowel	$3.7144E+05$	$3.5967E+05$	$3.8044E+05$	10	5
GA	Vowel	$1.7974E+05$	$1.5821E+05$	$1.9897E+05$	10	5
SMA	Vowel	$2.1442E+05$	$1.9209E+05$	$2.3157E+05$	10	5
HHO	Vowel	$4.7834E+05$	$3.8600E+05$	$5.5596E+05$	10	5
$HHO-SMA$	Vowel	$1.7810E+05$	$1.5808E+05$	$1.8881E+05$	10	5
ABC	Vowel	$3.4816E+05$	$2.5334E+05$	$3.5870E+05$	10	6
SSA	Vowel	$3.6060E+05$	$3.5130E+05$	$3.7156E+05$	10	6
GA	Vowel	$1.9961E+05$	$1.4517E+05$	$2.4911E+05$	10	6
SMA	Vowel	$1.9660E+05$	$1.7865E+05$	$2.2217E+05$	10	6
HHO	Vowel	$5.3413E+05$	$4.7725E+05$	$6.2863E+05$	10	6
$HHO-SMA$	Vowel	$1.7870E+05$	$1.5716E+05$	$2.1636E+05$	10	6
ABC	Wine	$2.5574E+04$	$2.3360E+04$	$2.5820E+04$	10	5
SSA	Wine	$3.6060E+04$	$3.5130E+04$	$3.7156E+04$	10	5
GA	Wine	$1.9964E+04$	$1.7502E+04$	$2.3848E+04$	10	5
SMA	Wine	$1.1267E+04$	$1.0786E+04$	$1.2249E+04$	10	5
HHO	Wine	$4.7362E+04$	$4.6380E+04$	$5.0410E+04$	10	5
$HHO-SMA$	Wine	$1.1793E+04$	$1.0569E+04$	$1.2564E+04$	10	5

Table 9: Comparison of Silhouette score.

Algorithm	Iris	Vowel	Wine	WDBC
$K-means$	0.68	0.42	0.57	0.69
$Hierarchical\,(Ward)$	0.67	0.44	0.58	0.65
$DBSCAN$	0.50	0.39	0.45	0.50
$Spectral\,Clustering$	0.70	0.46	0.60	0.69
$Gaussian\,Mixture\,(GMM)$	0.66	0.41	0.55	0.60
$HHO-SMA$	0.85	0.52	0.72	0.80

which requires more intricate clustering, respectively. With a score of 0.72, HHO-SMA showed a significant improvement over spectral clustering, which was around 20% better than spectral clustering, which had a superior score of 0.60.
With a score of 0.69, spectral clustering was the standard approach that performed the best on the sensitive WDBC dataset. Nevertheless, HHO-SMA performed noticeably better, scoring 0.80, which was more than 23% better than other conventional techniques and 16% better than spectral clustering.
In general on the straightforward Iris dataset, techniques like k-means and hierarchical clustering performed well; but, on the more complicated Vowel and Wine datasets, their performance significantly declined. Despite its capacity to detect non-linear clusters, DBSCAN performed poorly on all datasets. Spectral clustering and the GMM occasionally performed very well, but on all datasets, especially on the delicate and complicated WDBC and Wine, the HHO-SMA approach showed unquestionably better performance. The clusters produced by HHO-SMA are optimized in terms of both external separation (between clusters) and internal cohesion (within clusters), as indicated by the high Silhouette score. In fact, the results in the table demonstrate that the proposed HHO-SMA algorithm can effectively solve clustering problems in a variety of situations due to its flexibility and processing capacity that given its strong performance on increasingly complicated datasets like Wine, the proposed algorithm successfully solved the drawbacks of traditional approaches. Certainly the effective combination of two optimization algorithms in the proposed method enables the algorithm to perform global optimization and avoid local optima, leading to high-quality clustering and reduced cluster overlap.

4.4 Stability evaluation of proposed method via graphical presentations

In this part, we compare the performance of the proposed HHO-SMA method against the SMA and HHO algorithms for clustering in two scenarios: three clusters and five clusters on the Iris dataset. The convergence curves are displayed for the 5-cluster scenario in Figures 3 and 4, and for the 3-cluster scenario in Figures 5 and 6. The objective function's value, which is minimized in this instance, is represented by the y-axis, while the x-axis shows the number of iterations.
Figures 4 and 6 show that in both cases, HHO-SMA converges more quickly than SMA and HHO. The greater slope of the HHO-SMA curve at the start of the search phase makes this quite evident. In the 3-cluster case (Figure 6), for example, HHO-SMA takes about 50 iterations to get the near-optimal value, while SMA and HHO take more than 100 and 150 iterations, respectively. This pattern is also seen in the 5-cluster situation (Figure 4), with the distinction that for all three algorithms, the number of iterations needed for convergence rises, suggesting that the problem becomes more complex as the number of

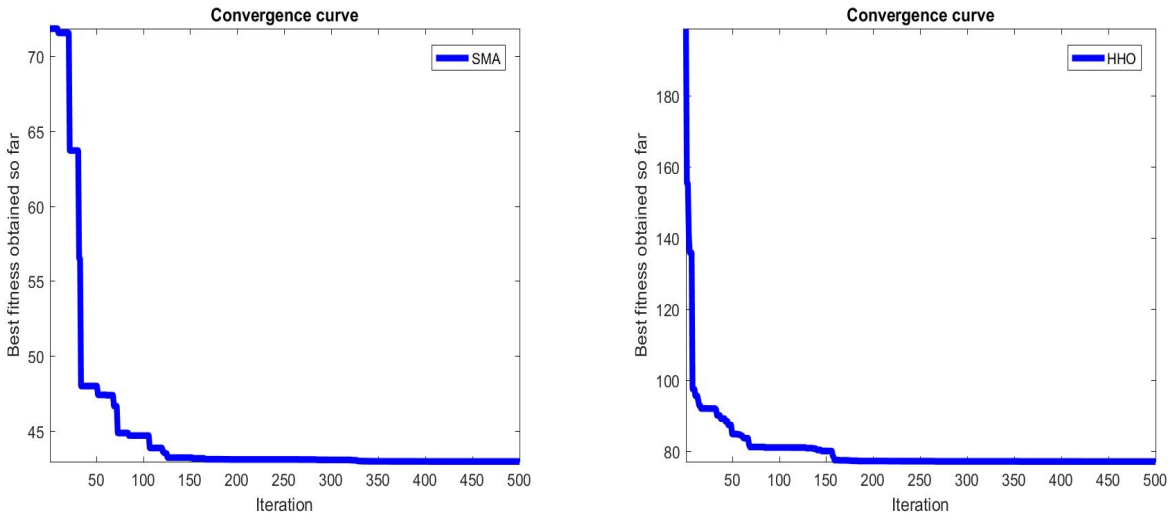


Figure 3: A: SMA's performance on the Iris with 5 clusters. B: HHO's performance on the Iris with 5 clusters.

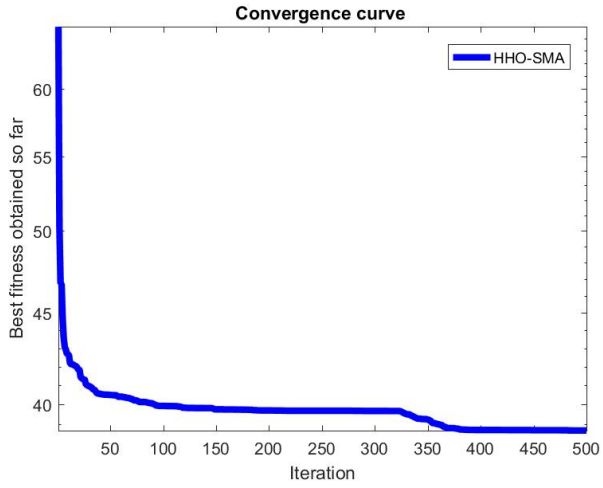
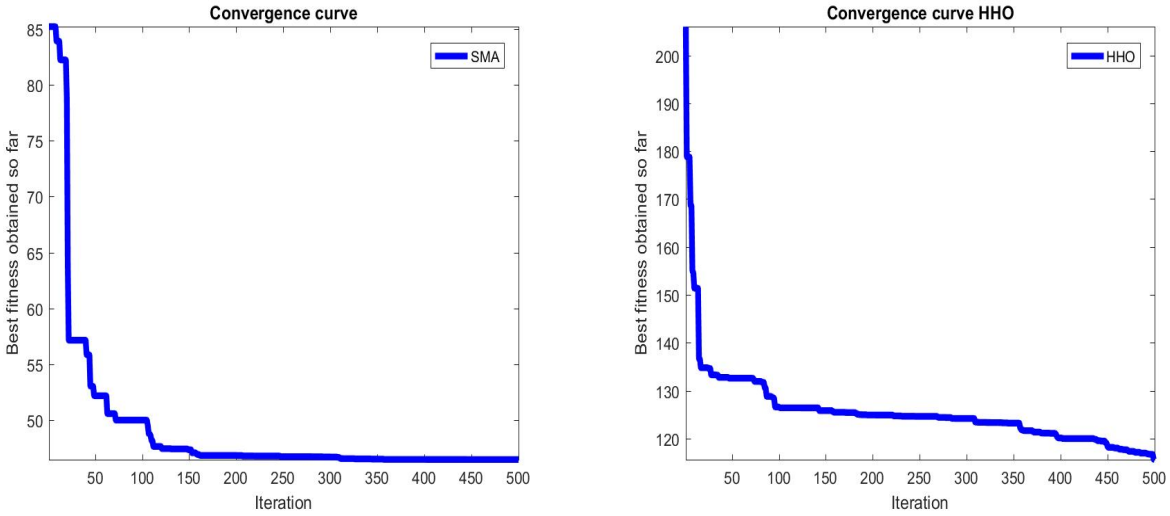


Figure 4: HHO-SMA's performance on the Iris with 5 clusters.



(C) (D)

Figure 5: C: SMA's performance on the Iris with 3 clusters. D: HHO's performance on the Iris with 3 clusters.

clusters increases.

Beyond convergence speed, HHO-SMA achieves a lower (better) objective function value than the other two algorithms in both scenarios. This demonstrates that HHO-SMA is capable of finding more optimal or near-optimal solutions compared to SMA and HHO.

The stability of the algorithms was also examined by analyzing the fluctuations of the convergence curves after reaching the near-optimal value. The results show that HHO-SMA exhibits greater stability than SMA and HHO in both scenarios. Overall, the experimental findings show that HHO-SMA clusters the Iris dataset more effectively than SMA and HHO in both the 3-cluster and 5-cluster scenarios. Convergence speed, ultimate solution quality, and algorithm stability all demonstrate this superiority. The reason for this is that HHO-SMA effectively combines SMA and HHO features, improving the balance between exploration and exploitation in the search space.

4.5 Implications and applications

The findings displayed show how well the proposed method performs on a variety of optimization tasks. When completing simpler problems, the method retains great speed and stability while demonstrating robustness in addressing challenges related to multi-modal functions. The results discussed in the previous sections establish the proposed method as a dependable and effective remedy for a range of practical optimization situations, including:

1. Engineering design optimization: Including system parameter tuning and structural design.
2. Machine learning hyperparameter tuning: Enhancing neural networks' or other algorithms' model parameters.

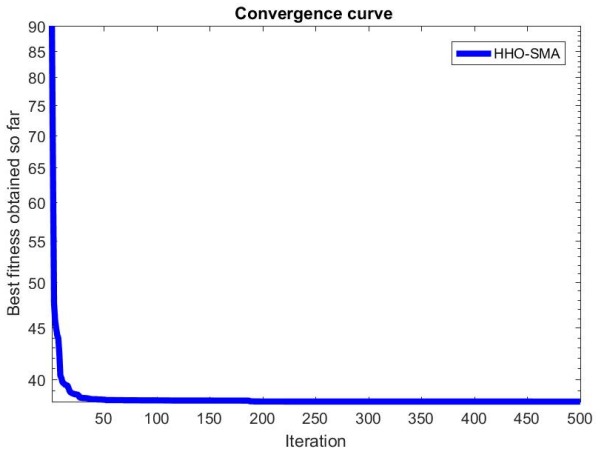


Figure 6: HHO-SMA's performance on the Iris with 3 clusters.

3. Robotics path planning: Determining the best paths for robotic systems in dynamic settings.
4. Supply chain optimization: Resolving issues with resource allocation, inventory, and logistics.
5. Financial portfolio management: Maximizing returns while minimizing risk through asset allocation optimization.

This adaptability and diversity highlight the usefulness of the proposed method in a variety of optimization scenarios.

5 Conclusion

Recently, hybrid methods have gained significant attention among researchers in combinatorial optimization and satisfactory results have been obtained. This study presents a novel hybrid algorithm. One of the basic challenges in the clustering discussion is to find optimal centers to avoid stagnation in local optima. In this article, according to the effective performance of metaheuristic methods in these problems, a hybrid method of two metaheuristic algorithms SMA and HHO, is proposed to find optimal centers for the clusters. The hybrid nature of these methods allows them to use the strengths of multiple algorithms to provide accurate and efficient results. In fact, this hybrid method offers promising solutions for clustering problems through the incorporation of computational intelligence techniques and mimicking natural behavior. According to the experiments and investigations, the outcomes demonstrated that the proposed strategy was operating as intended. In conclusion, the proposed HHO-SMA algorithm demonstrates superior performance across diverse optimization and clustering tasks by effectively combining the exploration strength of HHO with the exploitation capabilities of SMA. The results highlight its rapid convergence, remarkable stability, and robustness in tackling both unimodal and multimodal functions, as well as complex clustering problems. Compared to standalone algorithms and other benchmark methods, HHO-SMA constantly achieves lower fitness values, reduced variation across runs, and

enhanced solution quality, particularly in high-dimensional and challenging datasets. This underscores its adaptability, reliability, and potential for real-world applications in fields such as engineering optimization, machine learning, and data clustering. These findings validate HHO-SMA as a powerful and efficient optimization framework for addressing complex challenges in various domains.

Data availability statement

The datasets used in this paper are:
1- The Iris flowers dataset from https://www.kaggle.com/datasets/arshid/iris-flower-dataset
2- The Vowel dataset from https://www.kaggle.com/datasets/darubiano57/dataset-of-vowels
3-The Wine dataset from https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Abdollahzadeh, B., Gharehchopogh, F. S., Khodadadi, N., Mirjalili, S. (2022). Mountain gazelle optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems. Advances in Engineering Software, 174, 103282.

[2] Barkhordari Firozabadi, S., Shahzadeh Fazeli, S. A., Zarepour Ahmadabadi, J., Karbassi, S. M., (2023). Improving the performance of the FCM algorithm in clustering using the DBSCAN algorithm. Iranian Journal of Numerical Analysis and Optimization, 13(4): 763-774, https://doi: 10.22067/ijnao.2023.82361.1260.

[3] Bezdek, J. C., Ehrlich, R., Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. Computers and geosciences, 10(2-3), 191-203.

[4] Digalakis, J. G., Margaritis, K. G. (2001). On benchmarking functions for genetic algorithms. International journal of computer mathematics, 77(4), 481-506.

[5] Ester, M., Kriegel, H. P., Sander, J., Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. Inkdd, (34), 226-231.

[6] Faris, H., Mirjalili, S., Aljarah, I., Mafarja, M., Heidari, A. A. (2020). Salp swarm algorithm: theory, literature review and application in extreme learning machines. Nature-inspired optimizers: theories, literature reviews and applications, 185-199.

[7] Gandomi, A. H., Yang, X. S., Alavi, A. H. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Engineering with computers, 29, 17-35.

[8] 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.

[9] Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., Heming, J. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. Information Sciences, 622, 178-210.

[10] Ishak Boushaki, S., Kamel, N., Bendjeghaba, O. (2018). High-dimensional text datasets clustering algorithm based on cuckoo search and latent semantic indexing. Journal of Information and Knowledge Management, 17(3), 1850033.

[11] Jain, M., Saihjpal, V., Singh, N., Singh, S. B. (2022). An overview of variants and advancements of PSO algorithm. Applied Sciences, 12(17), 8392.

[12] Katoch, S., Chauhan, S. S., Kumar, V. (2021). A review on genetic algorithm: past, present, and future. Multimedia tools and applications, 80, 8091-8126.

[13] Kaur, S., Awasthi, L. K., Sangal, A. L., Dhiman, G. (2020). Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. Engineering Applications of Artificial Intelligence, 90, 103541.

[14] Kaveh, A., Talatahari, S., Khodadadi, N. (2020). Stochastic paint optimizer: theory and application in civil engineering. Engineering with Computers, 1-32.

[15] Kennedy, J., Eberhart, R. (1995). Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks IEEE, 4, 1942-1948.

[16] Khalilpourazari, S., Khalilpourazary, S. (2019). An efficient hybrid algorithm based on Water Cycle and Moth-Flame Optimization algorithms for solving numerical and constrained engineering optimization problems. Soft Computing, 23, 1699-722.

[17] Khanduja, N., Bhushan, B. (2020). Recent advances and application of metaheuristic algorithms: A survey (2014-2020). Metaheuristic and Evolutionary Computation: Algorithms and Applications, 207-228.

[18] Li, M. D., Zhao, H., Weng, X. W., Han, T. (2016). A novel nature-inspired algorithm for optimization: Virus colony search, Adv. Eng. Softw. 92, 65-88. https://doi.org/10.1016/j.advengsoft.2015.11.004.

[19] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, (14), 281-297.

[20] Mirjalili, S., Hashim, S. Z. M. (2010). A new hybrid PSOGSA algorithm for function optimization. 2010 international conference on computer and information application IEEE.

[21] Mirjalili, S., Mirjalili, S. M., Lewis, A. (2014). Grey wolf optimizer. Advances in engineering software, 69, 46-61.

[22] Mohammadi-Balani, A., Nayeri, M. D., Azar, A., Taghizadeh-Yazdi, M. (2021). Golden eagle optimizer: A nature-inspired metaheuristic algorithm. Computers and Industrial Engineering, 152, 107050.

[23] Mosavi, E., Shahzadeh Fazeli, S. A., Abbasi, E., Kaveh-Yazdy, F., (2025). Using Hybrid Mountain Gazelle Optimization and Particle Swarm Optimization Algorithms to Improve Clustering. Cluster Comput., https://doi.org/10.1007/s10586-024-05029-7.

[24] Mosavi, E., Shahzadeh Fazeli, S. A., Abbasi, E., Kaveh-Yazdy, F., (2025). Unite and Conquer Approach for Data Clustering Based on Particle Swarm Optimization and Moth Flame Optimization. Iranian Journal of Numerical Analysis and Optimization, https://doi:10.22067/ijnao.2025.89754.1508.

[25] Oyewole, G. J., Thopil, G. A. (2023). Data clustering: application and trends. Artificial Intelligence Review 56(7), 6439-6475.

[26] Price, K. V. (2013). Differential evolution. Handbook of optimization: From classical to modern approach, Berlin, Heidelberg: Springer Berlin Heidelberg, 187-214.

[27] Purushothaman, R., Rajagopalan, S. P., Dhandapani, G. (2020). Hybridizing Gray Wolf Optimization (GWO) with Grasshopper Optimization Algorithm (GOA) for text feature selection and clustering. Applied Soft Computing, 96, 106651.

[28] Rajwar, K., Deep, K., Das, S. (2023). An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges. Artificial Intelligence Review, 56(11), 13187-13257.

[29] Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., Costa, L. D. F., Rodrigues, F. A. (2019). Clustering algorithms: A comparative approach. PloS one, 14(1), p.e0210236.

[30] Saremi, v, Mirjalili, S., Lewis, A. (2017). Grasshopper optimisation algorithm: theory and application, Advances in engineering software, 105, 30-47.

[31] Sarkar, S., Roy, A., Purkayastha, B. S. (2014). A comparative analysis of particle swarm optimization and K-means algorithm for text clustering using Nepali Wordnet. Int. J. Nat. Lang. Comput.(IJNLC), 3(3).

[32] Shimin, Li., Chen, H., Wang, M., Heidari, A. A., Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. Future Generation Computer Systems, 111, 300-323.

[33] Singh, T., Saxena, N., Khurana, M., Singh, D., Abdalla, V., Alshazly, H. (2021). Data clustering using moth-flame optimization algorithm. Sensors, 21(12), 4086.

[34] Talbi, E. G. (2002). A Taxonomy of Hybrid Metaheuristi. Journal of Heuristics, 8, 541-546.

[35] Talbi, E. G. (2009). Metaheuristics: from design to implementation. John Wiley ans Sons.

[36] Yang, X. S. (2010). Firefly algorithm, stochastic test functions and design optimisation. International journal of bio-inspired computation, 2(2), 78-84.

[37] Yao, X., Liu, Y., Lin, G. (1999). Evolutionary programming made faster. IEEE Trans Evol Comput, 3, 82–102.