Evolutionary Intelligence (2023) 16:133–151
https://doi.org/10.1007/s12065-021-00647-1

RESEARCH PAPER

Local neighbour spider monkey optimization algorithm for data clustering

Vaishali P. Patel[1] · Manoj Kumar Rawat[1] · Amit S. Patel[2]

Received: 25 July 2020 / Revised: 27 May 2021 / Accepted: 25 July 2021 / Published online: 8 August 2021

Abstract
Data clustering plays a crucial role in the analysis of information collected from a variety of domains. Researchers developed many classical and mathematical algorithms to solve real-life problems, but due to the inherent property of these algorithms, they prematurely converge and fall to local optima. A further pattern of data in terms of shape, size, and distribution has a significant effect on the exploitation and exploration characteristic of algorithms which draw attention to many researchers. This work attempts to solve this problem by proposing an LNSMO local neighbour spider monkey optimization algorithm for data clustering. In the proposed algorithm Local Leader Phase of the spider monkey optimization algorithm is improved with its neighbour solution. Further to enhance the global search global leader phase of spider monkey optimization is improved with a chaotic operator. The performance of LNSMO is compared with eleven real-life datasets with five well-known Meta-heuristic algorithms in terms of a sum of within-cluster distance and convergence speed. It is further compared with recently developed hybrid meta-heuristic algorithms. Experimental result demonstrates that the proposed algorithm provides a better result in terms of Accuracy, F-measure, and SWCD.

Keywords Spider monkey optimization · Swarm optimization · Neighbour search · Chaotic factor · Clustering

1 Introduction

The objective of the clustering method is to classified data into its respective class (Clusters) such that data having similar property falls to the same cluster and different clusters have different properties. The success of any clustering algorithm is purely dependent on how it has been designed, for instance, its encoding scheme, distance measure or objective function, data assignment technique, search strategy, etc. Clustering techniques are broadly defined in two domains Hierarchal and partition clustering [1]. Where former generate a tree-like structure and later divide data such that single data assign to only one cluster. Among them, partition clustering is the first choice for researchers in the past few decades. K means algorithms, which are based on proximity measure as a distance is more preferred as it is easy to implement and flexible for hybridization with other algorithms. But it has limitations with premature convergence and speed as it's depending on the initial condition of clusters [2].

To find a new pattern in living organisms and frame to mathematical steps is an emerging trend among data researchers. This trend is broadly defined in two subdomain evolutionary and swarm-based algorithms. Evolutionary algorithms are derived from the principle of natural evolution which is existing on this planet for the past millions of years. And follow the interactive pattern of personal growth, collective development, breed selection, and reproduction to survive on the planet. These algorithms are older and more mature. Genetic Algorithm (GA), Differential Evolution (DE), Genetic programming (GP), Genetic Improvement, (GI), Evolution strategy (ES), Linear Genetic Programming (LGP), Gene Expression Programming (GEP), etc. are well known EA algorithms [3].

Swarm-based algorithms are structure with the population having agents interacting local with neighbours and

✉ Amit S. Patel
aspatel.mh@ddu.ac.in

Vaishali P. Patel
vaishalipatel02@yahoo.com

Manoj Kumar Rawat
drmkrawat@gmail.com

[1] Department of Computer Engineering, Oriental University, Indore, Madhya Pradesh, India

[2] Department of Mechanical Engineering, Dharmsinh Desai University, Nadiad, Gujarat, India

with the global environment, decentralize and self-organized pattern exist in the foraging process. [4]. Examples of SI include Particle swarm optimizer (PSO), artificial bee colony (ABC) algorithm, glow-worm swarm algorithm (GSA), firefly algorithm (FFA), cuckoo search algorithm (CSA), bat algorithm (BA), grey wolf optimizer (GWO), Whale optimization algorithm (WOA), Spider Monkey Optimization (SMO) and so on [5].

In the application of Meta-heuristic algorithms, they have to process a huge amount of data may be neighbour or unknown region. And both should be effectively searched to obtain a true or near optimum solution. In literature, it is known as exploitation and exploration respectively. More weightage to search around local space lead to premature and fast convergence and lessen the effect of global solution on the opposite side more weightage to exploration lead to slow convergence and unpredicted result [5]. Therefore it is an open problem for the researcher to balance between two search spaces. In literature number of researchers tries to solve a said problem with a different approach of initialization, update strategy, proximity measure distance, and many more.

The main contribution of this paper is to improve the search process of the spider monkey optimization algorithm hybridizes with a local neighbour search. This algorithm uses SMO as a global search and to refined search space, the neighbour search is embedded with the local leader phase of SMO. To improve global search a nonlinear property as a chaotic factor is implemented to the global leader phase of the proposed algorithm.

The proposed LNSMO (local neighbour-based spider monkey optimization) algorithm attempts to balance between local and global search space while preserving diversity property. The working of the proposed LNSMO algorithm has been analyzed using the eleven datasets from the UCI repository [6] and is compared with five well-known clustering algorithms like Particle swarm optimization, Genetic Algorithm, Grey wolf optimization, Differential Evolution, and Spider monkey optimization. It is further compared with seven recently developed hybrid clustering algorithms like, VDEO [7], AMADE [8], PSOPC [5], GLPSOK [9], WOAC [10], KMCLUST [11], and TEABC_elite [12]

The performance of the LNSMO algorithm is evaluated based on the sum of within-cluster distance or intra-cluster distance as an objective function and based on convergence speed. Finally to validate the clustering result of the assignment of data to the respective class is correct or not, two quality measures Accuracy and F-measure based on the confusion matrix are calculated. To check the significance of the proposed algorithm with a competent algorithm, statically an unpaired t-test is performed on the clustering result.

The simulation result of the proposed algorithm shows that the LNSMO algorithm performs better than its competitors in terms of SWCD and convergence characteristics. The result of the quality measure indicates that the proposed algorithm is more efficient in terms of the assignment of data to the correct cluster. Analysis of the t-test shows that LNSMO is statically significant compare to a competent algorithm.

The rest of this paper is organized in the following sequence. Section 2 describes the basic definition of the clustering problem and followed by past work done in the field of meta heuristic-based clustering algorithms. Section 3 explains the basic steps of the spider monkey optimization algorithm. Section 4 presents the proposed algorithm with the neighbour search method and chaotic operator. The experimental data set, parameter setting, and results are presented in Sect. 5. Finally, the conclusion and future direction are presented in Sect. 6.

2 Theoretical background

This section first describes the mathematical background and property of the clustering method and thereafter a brief literature review.

2.1 Clustering theory

Data clustering is the method of dividing n number of data points into the finite number of clusters k based on some similarity property having d-dimensional space (attribute). Let the set of n data points with d-dimension be written as $Y = \{y_1, y_2, y_3..., y_n\}$. The k number of clusters can be written as $Z = \{z_1, z_2, z_3..., z_k\}$, such that data assign to the same cluster have the same property and different clusters have different properties. And cluster constrained to the following properties [13].

Each cluster should not empty and have at least one member or data point.

$$Z_i \neq \phi, \forall i \in \{1, 2, ..., k\} \tag{1}$$

Two different clusters should not have a common member or data point

$$Z_i \cap Z_j = \phi, \forall i \neq j \text{ and } i, j \in \{1, 2, ..., k\} \tag{2}$$

Each data point should belong to a cluster

$$\bigcup_{i=1}^{k} Z_i = Y \tag{3}$$

The clustering problem with the Meta-heuristic algorithm is an optimization of the data point's separation such that it

creates a well-separated compact cluster and also preserves the above properties. And for better optimization, it required an effective objective or fitness function. The fitness function also accounts for a measure of the partitioning of data points. Mean square error is the most commonly used fitness function in the clustering problem and is defined as [5].

$$f(Y,Z) = \sum_{j=1}^{n} \min\{||y_j - z_i||^2\}, \{i = 1,2,3,...k\} \quad (4)$$

where, min represents the minimum distance between data points y_j and cluster z_i or similarity measure. In this paper well known Euclidean distance is used as a similarity measure.

2.2 Related work

Researchers try to solve the clustering problem either by evolutional or swarm-based algorithms.

Bouyer [14] developed a hybrid clustering method based on improved cuckoo and modified particle swarm optimization with K -Harmonic Means, In this work velocity equation of particle, is updated with global worst and personal worst solutions to balance between local and global search. The advantage of this algorithm is a parameter of the cuckoo search algorithm is updated automatically. Lei Yang [15] presents PSO clustering which is formulated as a tree structure and neighborhood property. In PSO velocity is modified with four components, basic velocity, particle current, and its best position, particle current and population best, particle current, and neighbour best position to refined the search process. PSO is designed as a tree structure in which the tree structure is updated with iteration such that the parent node is better than the child node. Neetu Kushwaha [16] presents teaching–learning-based optimization embedded with PSO and applied to the clustering problem. In which output of TLBO is taken as an input parameter to PSO for fine refinement of clustering results. Yugal Kumar [17] presents cat swarm optimization improve by modification in governing equation and extended with clustering problem. To improve global search global best is embedded in the equation of tracking mode velocity. Acceleration parameters of the algorithm are made dynamic to balance the search process. It is compared with K-Means, genetic algorithm, particle swarm, teaching–learning based optimization, ant colony optimization, and cuckoo search optimization algorithm. And the result shows better performance compare to its competitor. J. Prakash [18] present ABC with the global best property for the clustering problem. In this algorithm to speed up and keep away premature convergence global best and crossover operator is combined with ABC. Results demonstrate that the developed algorithm outperforms ABC and its variants. Ibrahim Aljarah [19] developed Locality informed GWO and applied it to clustering analysis. In this algorithm performance of GWO is improved by the tabu search operator which is acts as a local search near the best solution. The addition of new terms refined search space and perform better than compared algorithms like K-Medoids, K-means, hierarchical clustering and, furthest first techniques. Krishna Gopal Dhal [20] presents a cuckoo search algorithm with modification to balance between local and global search strategy and applied to the clustering method. This exploration is improved by global best and mutation with Levy and Cauchy distribution. A further mutation is controlled by step size derived from fitness. Exploitation is improved by k-neighborhood and previous personal solutions. In the proposed work author tries to improve clustering results by new search strategies and dynamic parameters updating. Yuefeng Tang [21] proposed Glowworm swarm optimization improved by variable step size which is a function of the level of luciferin carried by each glowworm instead of fixes one and initialization method based on iterations. And an improved algorithm was applied to the clustering problem. Yating Li [22] proposed Chaotic starling PSO for clustering. In this algorithm, acceleration coefficients are updated with the logistic map and exponential function is used to update inertial weight. To avoid trapping into local optima, a dynamic disturbance term is added to the velocity equation. Further starling bird's local search capability and neighbours information is used to direct particle search direction. The main drawback algorithm is that it may trap local optima when applied to a problem with multiple local optima. Farzaneh Zabihi [23] present a history driven ABC algorithm to balance local, global search result and extended to data clustering. In this algorithm, a memory archive structure is used to store individual fitness and position which is useful for avoiding fitness calculation. A guided anisotropic search strategy is used to improve local search. Scout bees mutation is improved with the global best component. Ashish Kumar Tripathi [24] proposed grey wolf optimization and enhanced it with new search strategies further applied to data clustering. In this work hunting strategy of a grey wolf is improved with Lévy Flight and Binomial crossover for prey to improve the exploration and exploitation capabilities. It is further parallelized by the map-reduce model in the Hadoop framework for clustering a large number of data sets. Vijay Kumar [25] proposed a clustering approach using a grey wolf algorithm. This algorithm adapts the searchability of GWO to overcome the weakness of local optima in the K-means algorithm. Pranesh Das [11] proposed a modified bee colony optimization for data clustering. To improve clustering and convergence results untrustworthy bees are getting change to participate with some probability. From the second iteration onward center of the previous solution is considered as a center of a present cluster solution. Further to maintain diversity, data that are not assigned in the previous stage

are utilized. Hassanzadeh and Meybodi [26] applied the property of the firefly algorithm to K means for improvement in premature convergence of the K- means algorithm. Cluster centers are found by FA and further refined by K-means algorithm. The proposed algorithm is compared with k mean, PSO, KPSO, and proves to be better. Amol kumar [27] proposed a hybrid approach of the exponential grey wolf and whale optimization for clustering. In this approach, the hunting mechanism of the whales is utilized to find a number of clusters, and exploration is improved by exponential grey wolf optimization. Amr Mohamed [28] presents a modified step whale optimization algorithm hybridized with tabu search and applied to the clustering problem. In the proposed work diversification of whale optimization is improved by changing swarm location based on their original position. To preserve the best location from the search space memory element of tabu search is utilized. The further search process is improved by a crossover operator. Yaping Li [29] proposed an improved glow worm algorithm for clustering. For better clustering results good-point set theory is used to distribute the initial population of the K-means algorithm and result further optimized with glow worm optimization. Roselyn Isimeto [30] proposed a glow worm swarm algorithm in which the sensor range parameter of glow warm swarm is found by min, max value of the sensory range, and fitting to quadratic function and solved by a least square method. Further to improve the result, the sum of the mean squared error is made a function of iteration by multiplying the number of clusters that change with iteration dynamically. Manju Sharma [5] proposed polygamous crossover-based PSO for clustering. In the proposed algorithm polygamous crossover operator is embedded with the velocity equation of PSO to refined a search process. The proposed algorithm compared with PSO, GA, DE, FA, GWO, and prove to be better in terms of SWCD. Nehsat et al. [31] presented a hybrid approach of PSO to perform a global search with K- means to perform a local search. Saida et al. [32] applied a cuckoo search capability to the clustering problem. And show better results in terms of distance measure and convergence.

In the proposed work spider monkey optimization improved with neighbour search and a chaotic operator is proposed for data clustering. The next section represents the basic SMO with its limitation, neighbour search method, and chaotic operator to develop the proposed algorithm.

3 SMO and neighbour search method

This section describes the basic theory and limitation of SMO and neighbour search methods which inspired the author to develop a new algorithm.

3.1 Spider monkey optimization algorithm

Spider monkey optimization (SMO) algorithm is developed by Jagdish Chand Bansal [33]. It is derived based on the intelligence of spider monkey inspired by the social structure of FFSS (Fission fusion social structure). According to FFSS monkeys distribute them in different size groups for foraging and have the following characteristic. Initially, all spider monkeys were grouped into 40–50 individuals. Each group follows the common leader to find new food source terms as the global leader of that group. In case of lack of required quantity of food, the Global leader divides the main group into smaller groups each having three to eight members for forage. And each group is guided by a local leader. Local leaders decide on searching food sources in each sub-group separately. The member of the group uses unique sound as a communication for social bonding and defining boundaries for defense. The SMO algorithm is structured with six different phases to complete its forage process as discuses in subsequent steps.

In SMO initial population of N spider monkey is generated randomly. Let, D- Dimensional vector space denoted by Y_{ij} representing spider monkey with jth dimension and ith individual. Then Y_{ij} initialize as,

$$Y_{ij} = Y_{\min j} + U(0,1) \times (Y_{\max j} - Y_{\min j}) \tag{5}$$

where $Y_{\min j}$ and $Y_{\max j}$ represent the lower and upper limit of jth dimension of ith individual. *U(0,1)* is a random number between [0,1]. The subsequent section describes six phases of SMO.

3.1.1 LLP: Local leader phase

In this step new position of the individual is found by information of local leader and individual of the group by Eq. 6. To check the quality of solution fitness measure is used means if current fitness is better than the old solution then the position will be updated with a new value.

$$Ynew_{ij} = Y_{ij} + U(0,1) \times (LL_{kj} - Y_{ij}) + U(-1,1) \times (Y_{rj} - Y_{ij}) \tag{6}$$

where, LL_{kj} and Y_{rj} represent jth dimension of the local group leader and randomly selected rth spider monkey from kth group such that $r \neq i$.

To control the diversity in the current step perturbation rate Pr between [0.1, 0.8] is used, The Pseudocode of the local leader phase is given in Fig. 1.

3.1.2 GLP: Global leader phase

As shown in Eq. 7 position of all individuals is updated based on the position of a global leader and other members of the group.

for each member $Y_i \in k^{th}$ *group do*
  for each $j \in \{1,...D\}$ *do*
    If $\mathrm{U}(0,1) \geq \mathrm{Pr}$ *then*
      $Ynew_{ij} = Y_{ij} + U(0,1) \times (LL_{kj} - Y_{ij}) + U(-1,1) \times (Y_{rj} - Y_{ij})$
    else
      $Ynew_{ij} = Y_{ij}$
    end if
  end for
end for

Fig. 1 Pseudocode of local leader phase

$$Ynew_{ij} = Y_{ij} + U(0,1) \times (GL_j - Y_{ij}) + U(-1,1) \times (Y_{rj} - Y_{ij}) \tag{7}$$

where GL_j show the jth dimension of the global leader and $j \in \{1,...D\}$ is randomly chosen value. Further, the probability $prob_i$ which is a function of fitness as depicted in Eq. 8 is used to select a better candidate for the next iteration. Figure 2 Shows the Pseudocode of the global leader phase.

$$prob_i = \frac{fitness_i}{\sum_{i=1}^{N} fitness_i}$$

3.1.3 GLL: Global leader learning phase

In this step greedy selection based on the best fitness from a population is used to update the global leader position. If the global leader position is not updated then its global limit count is increment by 1.

3.1.4 LLL: Local leader learning phase

In this step greedy selection based on the best fitness from a particular group is used to update the local leader position.

Counter = 0;
while group size > counter ***do***
  for $\forall Y_i \in$ *group* ***do***
    If $\mathrm{U}(0,1) < prob_i$ ***then***
      Counter = counter + 1
      Choose $j \in \{1,...D\}$ *randomly*
      Choose $Y_r \in$ *group randomly with* $i \neq r$
      $Ynew_{ij} = Y_{ij} + U(0,1) \times (GL_j - Y_{ij}) + U(-1,1) \times (Y_{rj} - Y_{ij})$
    end if
  end for
end while

Fig. 2 Pseudocode of global leader phase

If the position of a local leader is not updated then its local limit count is increment by 1.

3.1.5 LLD: Local leader decision phase

This phase helps to decide the position of local leader group members. If any local leader position is not updated up to a predefined number of count (Local Leader Limit) then a member of its group position is generated either by random or by information gained from the local and global leader phase as per Eq. 9 With probability, *Pr*.

$$Ynew_{ij} = Y_{ij} + U(0,1) \times (GL_j - Y_{ij}) + U(0,1) \times (Y_{ij} - LL_{kj}) \tag{9}$$

It is revealed from Eq. 9 that the updated position of a spider monkey is repeal from local and attracted to global leader. Finally, the fitness of the updated position is calculated. Figure 3 Shows the Pseudocode of the LLD phase.

3.1.6 GLD: Global leader decision phase

This phase is the same as the previous but applied to the global leader phase. If the position of global leader is not updated for a predefined number of count (Global Leader Limit) then the global leader divides the population into small groups start with two and increment with one till a maximum number of group (MG) forms. Each time in the GLD phase local leader learning phase is repeated to elect a local leader with newly generated group positions. Even after dividing to a maximum number of the group if the position of global leader is not updated then the global leader merges all groups to form a single group.

Figure 4 shows the Pseudocode of the Spider monkey optimization algorithm.

3.2 Neighbour search method

In this method, a neighbour solution is found by moving the present solution by perturbation by some defined strategy. It is also known as neighbour search and result in a Local solution rather than a global solution. This method follows the

If *Local Leader Limit < Local Limit Count* ***then***
  Local Limit Count = 0
  for *each* $j \in \{1,...D\}$ *do*
    If $\mathrm{U}(0,1) \geq \mathrm{Pr}$ ***then***
      $Ynew_{ij} = Y_{\min j} + U(0,1) \times (Y_{\max j} - Y_{\min j})$
    else
      $Ynew_{ij} = Y_{ij} + U(0,1) \times (GL_j - Y_{ij}) + U(0,1) \times (Y_{ij} - LL_{kj})$
    end if
  end for
end if

Fig. 3 Pseudo code of LLD phase

Initialization: SM Population, Probability (*Pr*), Parameter: Local leader limit, Global leader limit
Evaluate fitness or objective function
Find global and local leaders by greedy selection.
While stopping criterion not met ***do***

1) Update the position of spider monkeys based on the LLP algorithm (Fig.1). It is the function of Previous position, local leader, and random *SM* position from the group.
2) Apply greedy selection between old and newly generated ***SM*** positions based on fitness.
3) Apply Eq.8 to find $prob_i$ for all group members.
4) Update all group member positions selected by $prob_i$ by GLP algorithm (Fig.2) which is a function of Previous, global leader, and random SM position from the group.
5) Apply global and local leader learning phase (Section 3.1.3, 3.1.4) with greedy selection to update local and global leader's positions on all groups.
6) If the position of any local group leader is not updated for a predefined number of counts (Local Leader Limit), then members of that group updated by the LLD phase algorithm (Fig.3).
7) If the position of global leader is not updated for a predefined number of counts (Global Leader Limit) then apply the global leader decision phase (Section 3.1.6) to split or merge the groups.

end while

Fig. 4 Pseudo code of spider monkey optimization

```
Initial solution: S;
Calculate fitness: f(S);
d=1;
while(d=1)
        d=0;
        for all candidate
                Select neighbour S* by neighbour search strategy
                Calculate fitness: f(S*);
                        if (f(S*)<f(S))
                        S ←S*
                        f(S) ←f(S*)
                        d =1;
                        end if
        end for
end while
```

Fig. 5 Pseudo code of neighbour search method

following basic procedure. [34]. The Pseudocode of Neighbour search method is given in Fig. 5.

Neighbourhood structure: Neighbour can be defined in many ways. In this work following strategies are applied [35].

Swap neighbourhood: Exchange of ith and jth position.

Insert neighbourhood: Remove the ith position and put it to the jth position.

Reverse neighbourhood: Select two random positions and reverse order of positions between selected positions.

In the SMO algorithm, the position of the spider monkeys is updated based on the positions of other randomly selected monkeys without checking its property. This leads to slow convergence, high breaking, and merging of groups [36]. Further, SMO has not the neighbour search property around the find solution. So to solve the described problem we have proposed, SMO with neighbour search embedded to Local Leader Phase of spider monkey algorithm. Further to improved global search, nonlinear property in terms of a chaotic operator is applied to the global leader phase of SMO.

4 The proposed algorithm (LNSMO)

To balanced exploitation and exploration following modifications are made in a basic variant of the spider monkey optimization algorithm. To improve the local search of an SMO algorithm, a neighbour search is embedded in the LLP phase of the SMO algorithm. In basic SMO, local leader search is influenced by randomly selected spider monkeys without checking its domination in terms of the objective function value. This shows that still there is a possibility of a better neighbour solution near to selected spider monkey. This encourages us to search neighbour solutions for better optimum results. Further to improve global search a chaotic operator derived from a logistic map is introduced in the global leader phase of SMO to overcome premature convergence. The flow chart in Fig. 6 shows the detailed steps of LNSMO and the Pseudocode in Fig. 7 shows the clustering application of the proposed algorithm. The descriptions of other steps of LNSMO are described below.

4.1 Neighbour search procedure

To find a better neighbour position in this work we applied swap, insert, and reverse search strategies [35] randomly to the local leader phase of SMO. Following pseudo code in Fig. 8 shows the modification in the Local leader phase with neighbour search procedure.

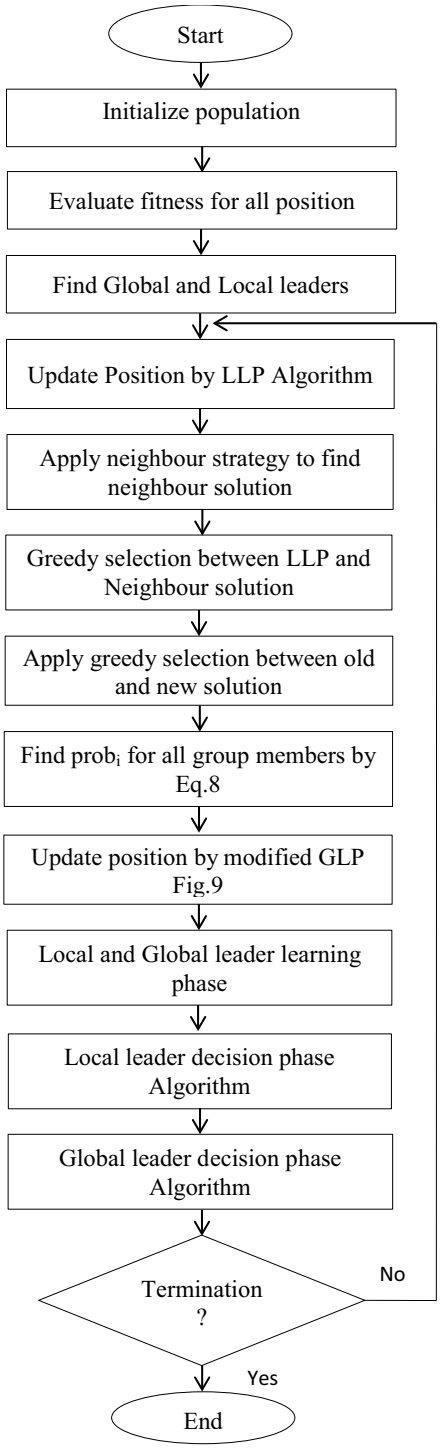

Fig. 6 Flow chart code of LNSMO algorithm

4.2 Modified global leader phase

Chaos is a phenomenon of nonlinear dynamics having a property of stochastic randomness, regularity, ergodicity, and sensitivity to an initial condition. This characteristic is helpful to avoid sticking to the local optimum in the search process [37, 38]. The main objective of this section is to improved global search by random and nonlinear property of chaotic factor using a logistic map as shown in Eq. 10

$$w = 0.5 \times rand() + 0.5 \times z \tag{10}$$

where, rand() is a random number between 0 to 1, And z is a logistic mapping given by Eq. 11

$$z = 4 \times z \times (1 - z) \tag{11}$$

where z is a random number between 0 to 1.

Finally, the position update equation of global leader phase is modified as,

$$Ynew_{ij} = Y_{ij} + w \times (GL_j - Y_{ij}) + U(-1, 1) \times (Y_{rj} - Y_{ij}) \tag{12}$$

Figure 9 shows the pseudo-code of the Modified global leader phase.

4.3 Population representation

Every SM position is represented as a search agent having a string of real numbers encoded with K number of cluster centers. The length of each position string is represented as K × D two-dimensional matrixes. Where D represents D dimensional search space. In this representation, each row specifies the cluster center and the column specifies the attribute of the input data set.

For example: Let K = 3 and D = 4, Problem with four attributes and number of cluster three then each position is represented as,

1.2	0.2	1.8	2	Cluster 1
1.3	0.4	2.1	1.9	Cluster 2
0.9	0.2	2	1.7	Cluster 3

4.4 Population initialization

For N number of population, each SM position is initialized with K randomly generated data points based on the minimum and maximum attribute value of the input data set as a cluster center.

4.5 Fitness function

Clustering with a meta-heuristic or swarm algorithm is an optimization problem and to solve the optimization problem we required an objective or fitness function. In this algorithm first Euclidian distance between every data point and cluster centers is calculated, and assign to the nearest cluster. Then after the center of the cluster is refined by the mean value of all data points belongs to that cluster and cluster center. Then fitness function sum of within-cluster distance (SWCD) is calculated

Input parameters:

Population size: N
Maximum iterations: Maxiter
Probability: Pr
Maximum local leader limit: MLL
Maximum Global leader limit: GLL
Maximum clusters: K
Population dimensions: D
Data set: X

Output: Cluster centers, SWCD

Begin

Generate initial position (SM) of N spider monkeys with K randomly selected clusters.
Find the Euclidian distance of every data point from the cluster center and assign it to the nearest cluster.
Find the fitness of every SM position by Eq.13.
Find global (GL) and local leaders (LL) by greedy selection.

for iteration = 1 to Maxiter

1) Update each SM position based on the local leader phase (Fig.8)
SM = LLP (SM, LL, Parameters, X);

a) Calculate the fitness of the newly generated SM position by Eq.13.
b) Apply neighbour search steps to SM position updated by LLP, and find a neighbour position.
c) Calculate the fitness of the neighbour position by Eq.13.
d) Update SM position by greedy selection between position updated by LLP and neighbour position generated by neighbour search steps.

2) Apply greedy selection between old and newly generated SM positions based on fitness.
3) Apply Eq.8 to find $prob_i$ for all SM positions.
4) Update SM position selected by $prob_i$ with the modified global leader phase (Fig.9).

SM = GLP (SM, GL, Parameters, X);

5) Apply global and local leader learning phases (Section 3.1.3, 3.1.4) with greedy selection to update local and global leader's positions on all groups.

LL = LLL (SM, LL, Parameters);
GL = GLL (GL, LL, Parameters);

Best Solution = GL;

6) If the position of any local group leader is not updated for a predefined number of counts (Local Leader Limit) then members of that group updated by the Local leader decision phase (Fig.3).

[SM, LL] = LLD (SM, GL, LL, Parameters, X);

7) If the position of global leader is not updated for a predefined number of counts (Global Leader Limit) then apply the global leader decision phase (Section 3.1.6) to split or merge the groups.

[GL, LL] = GLD (SM, GL, LL, Parameters);

end for

end

Fig. 7 Pseudo code of LNSMO clustering algorithm

for *each member,* $Y_i \in k^{th}$ *group do*
for *each* $j \in \{1,...D\}$ *do*
If $\mathrm{U}(0,1) \geq \mathrm{Pr}$ *then*

$$Ynew_{ij} = Y_{ij} + U(0,1) \times (LL_{kj} - Y_{ij}) + U(-1,1) \times (Y_{rj} - Y_{ij})$$

Calculate fitness: $f(Ynew_{ij})$

$$P = U(0,1) \times (LL_{kj} - Y_{ij}) + U(-1,1) \times (Y_{rj} - Y_{ij})$$

Select any two random positions from P say A and B, where $A \neq B$
Generate random integer number between (1, 3) = Z

switch *Z*
case 1
P = Swap (P, A, B);
case 2
P = Reversion (P, A, B);
case 3
P = Insertion (P, A, B);
end switch

$$Ynh_{ij} = Y_{ij} + P$$

Calculate fitness: $f(Ynh_{ij})$

if $f(Ynh_{ij}) < f(Ynew_{ij})$

$$Ynew_{ij} = Ynh_{ij}$$

$$f(Ynew_{ij}) = f(Ynh_{ij})$$

end if
else

$$Ynew_{ij} = Y_{ij}$$

end *if*

end *for*
end *for*

Fig. 8 Pseudo code of modified LLP with Neighbour search algorithm

Counter = 0;
while group size > counter **do**
for $\forall Y_i \in$ *group* **do**
If $\mathrm{U}(0,1) < prob_i$ **then**
Counter = counter + 1
Choose $j \in \{1,...D\}$ *randomly*
Choose $Y_r \in$ *group randomly with* $i \neq r$

$$Ynew_{ij} = Y_{ij} + w \times (GL_j - Y_{ij}) + U(-1,1) \times (Y_{rj} - Y_{ij})$$

end if
end for
end while

Fig. 9 Pseudo code of modified global leader phase

as an intra-cluster distance between data points and the cluster center to which they belong. For better clustering results minimum of SWCD is preferred and calculated by Eq. 13.

$$f(C_1, C_1...C_K) = \sum_{i=1}^{K} \sum_{X_j \in C_i} ||P_i - X_j|| \tag{13}$$

Here, P_i represents cluster center of C_i and X_j represents data belongs to cluster C_i.

4.6 Termination criteria

This algorithm runs for a predefined number of iterations and the solution found at the last iteration gives optimum cluster center.

5 Algorithm implementation

The proposed algorithm was implemented with MATLAB and compared with a well-known meta-heuristic algorithm as DE [39], PSO [40], GA [41], GWO [25], and SMO [33].

5.1 Data set

To perform experiments by the proposed algorithm author selects eleven data sets from the UCI repository [6] and the properties of these data sets are given in Table 1. The initial parameters of comparative algorithms are given in Table 2.

Table 1 Properties (Data set)

Name	Instances	Attributes	Classes
Glass	214	9	6
Cancer	283	9	2
Wine	178	13	3
Seed	210	7	3
Bupa	345	6	2
CMC	1473	9	3
Iris	150	4	3
Heart	270	13	2
Magic	19,020	10	2
HTRU2	17,898	8	2
Haberman	306	3	2

5.2 Quality measurements

To analyze the clustering results of the proposed algorithm following quality, measures are calculated [9]. These measurements show that the prediction of assigned data points to a particular cluster is correct as per true class or not. Larger the value better the clustering assignment quality.

Accuracy: It is the fraction of correctly classified to the total number of samples.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

F- measure which account precision and recall written as,

$$F - Measure = \frac{2 \times precision \times recall}{precision + recall} \tag{15}$$

where, TP = True Positive, TN = True negative, FP = False positive and FN = False negative samples.

Further intra-cluster distance in terms of SWCD Eq. 13 is calculated and compared with other algorithms.

5.3 Experiment results

5.3.1 Results and discussion based on SWCD values

Table 3 shows the best, mean, and standard deviation of SWCD values obtain by different algorithms over 20 independent runs. The objective of this experiment is to check whether the proposed algorithm capable of produced compact clustering results or not. To produce stable results each algorithm is repeated with 200 iterations for 20 runs. The SWCD results calculated here are the best, mean, and standard deviation for 20 independent runs calculated for the last or 200th iteration. In Table 3 bold text shows the best results compare to competent algorithm for SWCD value.

In the Glass dataset proposed algorithm end with a minimum value of SWCD compare to all algorithms, and gives the optimum of mean SWCD, but suffers little with standard

Table 2 Parameter setting

LNSMO Population size(N): 30 Max. Iterations:200 Local leader limit: D × N Global leader limit ∈ [N/2, 2 × N]: N Perturbation rate(Pr): 0.1	PSO Population size: 30 Max. Iterations:200 Inertia weight:1 C1 (Personal), C2(Global) learning coefficient:2	DE Population size: 30 Max. Iterations:200 Lower Bound (Scaling Factor):0.2 Upper Bound (Scaling Factor): 0.8 Crossover probability: 0.2
GA Population size: 30 Max. Iterations:200 Crossover probability:0.8 Mutation Probability:0.01	GWO Population size: 30 Max. Iterations:200 a: Decreasing 2 to 0 linearly	SMO Population size(N): 30 Max. Iterations:200 Local leader limit: D × N Global leader limit ∈ [N/2, 2 × N]: N Perturbation rate(Pr): 0.1

Table 3 Best, mean, and standard deviation of SWCD and time value obtained by different algorithms

	SWCD	GLASS	CANCER	Wine	Seed	Bupa	CMC	Iris	Heart	Magic	HTRU2	Haberman
PSO	Best	214.8213	2983.6237	16,510.8168	312.7844	9977.0098	5541.9213	96.9085	11,911.7977	1.6494E+06	1.1259E+06	2622.7435
	Mean	217.4966	2983.9380	16,513.8364	313.0065	9979.3308	5543.0723	96.9976	11,915.6180	1.6495E+06	1.1260E+06	2622.8785
	Std	**1.4484**	0.4418	7.3630	0.2153	1.3849	0.8410	0.0479	2.0781	9.7807E+01	6.4882E+01	0.5040
	Time	90.1207	69.6810	58.2915	54.2038	50.7010	142.8157	52.8382	60.4854	9.9792E+02	7.1052E+02	48.6642
GWO	Best	215.9894	2983.6277	16,508.6962	312.7765	9969.9286	5542.3863	96.9230	11,908.5794	1.6493E+06	1.1258E+06	2622.7476
	Mean	219.6321	2983.7202	16,511.1000	312.8539	9973.5622	5543.7448	97.0449	11,912.1398	1.6494E+06	1.1260E+06	2623.4625
	Std	2.8306	0.1164	3.1659	0.0613	2.2685	1.5110	0.0847	3.8148	**9.0872E+01**	8.3283E+01	0.5963
	Time	132.6804	73.2355	87.1231	82.9313	53.2514	132.0025	56.1983	71.0330	1.1108E+03	8.0147E+02	44.2724
GA	Best	214.2432	2983.6124	16,508.2543	312.7590	9974.7472	5541.3845	96.9317	11,916.7389	1.6493E+06	1.1258E+06	2622.7455
	Mean	219.9613	2984.1636	16,519.2669	312.9336	9977.1760	5542.6045	97.0529	11,922.6318	1.6496E+06	1.1260E+06	2624.9948
	Std	7.4772	0.4087	10.3521	0.2737	2.3462	1.4903	0.0971	3.8041	4.0198E+02	7.5869E+01	2.2212
	Time	66.9606	55.2414	42.7054	44.8467	39.0077	94.5006	39.9124	42.7763	7.5888E+02	5.7940E+02	37.5386
SMO	Best	233.9304	2965.1328	16,298.4893	312.0686	9851.7252	5538.6901	**96.5405**	11,852.9479	1.6268E+06	1.0641E+06	2566.9889
	Mean	262.1774	3009.5991	16,304.0704	315.4984	9876.7266	5588.8177	96.8937	11,862.9358	1.6859E+06	1.0798E+06	2567.1144
	Std	11.8517	47.7847	4.4506	4.4342	36.7252	41.5654	0.3532	11.8314	4.7262E+04	1.4325E+04	0.3062
	Time	126.9410	133.6412	119.3677	114.8908	115.5882	197.9177	112.2250	120.8128	1.2463E+03	1.1197E+03	111.2999
DE	Best	255.1448	2964.4441	16,301.3278	318.3963	9851.7226	5548.0985	97.8460	11,849.5751	1.6251E+06	**1.0635E+06**	**2566.9889**
	Mean	287.5873	2981.5492	16,314.0227	328.0744	9856.6339	5578.9967	100.4423	11,852.3689	1.6290E+06	1.0638E+06	2566.9945
	Std	13.2892	11.7474	5.6310	4.9805	6.3332	15.0819	1.5385	2.2596	2.4324E+03	3.1385E+02	0.0089
	Time	57.0044	56.9390	46.6900	48.6506	47.6146	77.9469	46.8168	51.9746	5.1360E+02	4.4556E+02	48.2381
LNSMO	Best	**210.4644**	**2964.3870**	**16,292.1847**	**311.7978**	**9851.7210**	**5532.1847**	96.6555	**11,849.4412**	**1.6230E+06**	**1.0635E+06**	**2566.9889**
	Mean	**212.7679**	**2964.3870**	**16,292.4381**	**311.7978**	**9851.7210**	**5532.1847**	**96.6555**	**11,849.4412**	**1.6265E+06**	**1.0635E+06**	**2566.9889**
	Std	2.1203	**4.70E-09**	**0.4660**	**1.06E-08**	**4.60E-09**	**7.63E-06**	**4.36E-09**	**6.49E-07**	2.4252E+03	**1.3536E+01**	**3.04E-09**
	Time	149.7854	155.5620	131.3254	125.1315	133.0992	232.3502	137.5315	135.7030	1.4291E+03	1.1891E+03	120.0262

deviation. For the cancer dataset LNSMO produced a minimum of SWCD which is comparable with DE, but a significant difference in mean value and standard deviation, which is nearer to zero. This shows the stability of the algorithm compares to others. For the Wine dataset LNSMO end with a more compact result compare to competent algorithms in best, mean, and standard deviation of SWCD. For Seed data output results are comparable with all algorithms, and better than DE. While standard deviation of LNSMO is nearer to zero and shows the higher stability of the proposed algorithm. The results of the Bupa dataset show that the proposed algorithm is comparable with DE but with higher stability and for other algorithms it shows a significant difference. In the CMC dataset LNSMO produced a significant difference in all measures with too small or near to zero variation in SWCD. In the iris dataset, LNSMO is comparable with all the competent algorithms but strong in standard deviation. For the Heart, dataset proposed algorithm produced a significant difference in all measures for all the algorithms, and comparable results with DE but a major difference in standard deviation. The bigger dataset of the Magic, the proposed algorithm produced better results in almost all measures for all the algorithms and comparable with DE. In the HTRU2 dataset, LNSMO produced comparable results with DE but a significant difference in standard deviation. In the Haberman, dataset LNSMO produced comparable results with DE but the more stable result with a minimum of standard deviation. In most cases, LNSMO produced a minimum value of SWCD with a small value of standard deviation which is nearer to zero, which shows the higher stability of the proposed algorithm against others.

5.3.2 Result and discussion based on cluster qualities measures

Table 4 shows the analysis of cluster quality in terms of accuracy and F-measure. The objective of these simulations is to check the assignment of data points to a cluster is as per the actual class or not. The larger value of these measures is preferred for better clustering results. These results are derived from the simulation of 20 independent runs of each algorithm and each algorithm is repeated for 200 iterations. In Table 4 bold text shows the best results compare to competent algorithm for accuracy and F-measure.

For the Glass dataset, the accuracy of LNSMO is second best with a small difference to GA which produces the best accuracy among all algorithms. However, the F-Measure value for the proposed algorithm is highest in all comparable

Table 4 Mean and standard deviation of cluster quality measure for datasets

	Measure	GLASS	CANCER	Wine	Seed	Bupa	CMC	Iris	Heart	Magic	HTRU2	Haberman
PSO	Accuracy	0.7141	0.9368	0.7256	0.5665	0.5004	0.5495	0.8438	0.5008	0.4743	0.4665	0.4941
		0.0497	0.4397	0.1757	0.2330	0.0088	0.0390	0.2047	0.0914	0.1159	0.1775	0.0192
	F Measure	0.6001	0.9344	0.5956	0.3076	0.4376	0.3214	0.8546	0.4964	0.4825	0.3374	0.4653
		0.1098	0.4381	0.2466	0.2894	0.0222	0.0591	0.3086	0.0881	0.0888	0.0728	0.0178
GWO	Accuracy	0.7168	0.9405	0.7031	0.5337	0.4990	0.5368	0.8622	0.5010	0.4992	0.4492	0.4902
		0.0325	0.4578	0.1788	0.1948	0.0059	0.0450	0.1814	0.0915	0.1206	0.1760	0.0174
	F Measure	0.6050	0.9383	0.5669	0.2971	0.4450	0.2986	0.8793	0.5005	0.4995	0.3303	0.4616
		0.0568	0.4562	0.2477	0.2919	0.0253	0.0688	0.2717	0.0881	0.0926	0.0721	0.0161
GA	Accuracy	**0.7276**	0.9201	**0.7736**	0.5643	0.5010	0.5445	0.8912	0.5003	0.4599	0.4492	0.5001
		0.0510	0.4573	0.1661	0.1824	0.0078	0.0429	0.1868	0.0915	0.1209	0.1767	0.0184
	F Measure	0.6010	0.9389	0.6710	0.3452	0.4343	0.3085	**0.9000**	0.4959	0.4759	0.3302	0.4720
		0.1070	0.4557	0.2240	0.2736	0.0236	0.0650	0.2800	0.0882	0.0922	0.0723	0.0171
SMO	Accuracy	0.7150	0.9295	0.7261	0.5998	0.5004	0.5522	**0.9009**	0.4998	0.4923	0.4488	0.5000
		0.0461	0.4567	0.1721	0.1529	0.0080	0.0408	0.1690	0.0920	0.1195	0.1761	0.0201
	F Measure	0.6001	0.9283	0.6078	0.3967	0.4401	0.3203	0.8967	0.4955	0.4958	0.3301	0.4707
		0.1023	0.4550	0.2347	0.2301	0.0236	0.0613	0.2532	0.0887	0.0913	0.0722	0.0186
DE	Accuracy	0.7167	0.9368	0.7489	0.5352	0.5003	0.5423	0.8896	0.4997	0.4883	0.4478	0.4941
		0.0474	0.4741	0.1479	0.2421	0.0056	0.0404	0.1443	0.0891	0.1022	0.1252	0.0192
	F Measure	0.6005	0.9347	0.6240	0.3014	0.4392	0.3048	0.8910	0.4955	0.4924	0.3341	0.4653
		0.0700	0.4722	0.2213	0.3630	0.0267	0.0611	0.2169	0.0859	0.0785	0.0495	0.0178
LNSMO	Accuracy	0.7182	**0.9649**	0.7211	**0.6562**	**0.5082**	**0.5537**	**0.9000**	**0.5087**	**0.5078**	**0.4923**	**0.5088**
		0.0463	0.0047	0.0159	0.1636	0.0053	0.0241	0.0420	0.0893	0.0737	0.1244	0.0200
	F Measure	**0.6104**	**0.9444**	**0.6997**	**0.4863**	**0.4998**	**0.3224**	**0.9006**	**0.5042**	**0.5060**	**0.5056**	**0.4725**
		0.0844	0.0473	0.0212	0.2456	0.0256	0.0380	0.0205	0.0861	0.0591	**0.0062**	0.0186

algorithms. For the Cancer dataset, LNSMO produced a significant difference in Accuracy which is highest among all comparative algorithms with smaller standard deviation. For F-measure proposed algorithm is comparable with others but smaller standard deviation. For the Wine dataset GA algorithm produced a significant difference in accuracy with LNSMO and other algorithms, but LNSMO dominated in F-measure with GA and other algorithms. Further proposed algorithm shows a smaller standard deviation in accuracy and F-measure compare to all the algorithms. In the Seed dataset, LNSMO output a higher difference in accuracy and F-measure compare to all the algorithms. In the Bupa dataset performance of LNSMO is similar to competitive algorithms accuracy, but the higher difference in F-measure. For the CMC dataset LNSMO produced similar accuracy results with competent algorithms. In F-measure results, LNSMO carries better results than GWO, GA, DE, and similar results with PSO and SMO. For the Iris dataset LNSMO produced better results than PSO, GWO and similar results compare to others in accuracy and F-measure. For the heart dataset proposed algorithm produced similar results compared to all algorithms in accuracy and F-measure. In the Magic dataset, LNSMO produced better results than PSO, GA, DE, and similar to the remaining algorithms for accuracy. In F-measure results, LNSMO produced better results than GA, PSO and similar to remaining algorithms. For the HTRU2 dataset proposed algorithm produced a significant difference in accuracy and F-measure compare to all algorithms. The standard deviation of the F-measure is small and nearer to zero which is best among all the algorithms. In the Haberman dataset, LNSMO produced similar results to all algorithms. Overall results show that the proposed algorithm predicts better or similar results compared to competent algorithms.

5.3.3 Convergence results

Figure 10 depicts the convergence plots for all eleven datasets between iterations and SWCD results. Overall results from Fig. 10 show that DE takes more iteration to converge. SMO algorithm does not trap to local search space but gives sub-optimum results. LNSMO predicts the minimum value of SWCD in less iteration and other algorithms converge prematurely with a higher value of SWCD. Figure 10a for the glass dataset reveals that the proposed algorithm makes a clearer boundary between premature algorithms and DE which take more iterations to converge. SMO does not fall to the local optimum but gives a high value of SWCD. Indirectly it shows that our algorithm is capable to balance between exploration and exploitation characteristics. Figure 10b depicts the convergence results for Cancer clearly shows that LNSMO takes half of the iterations to converge compare to DE. From Fig. 10c of wine, (d) seed, (e) Bupa, and (f) CMC data proposed algorithm converge faster than DE with a smaller value of SWCD, While SMO reveal the same convergence pattern of LNSMO but with higher value of SWCD. From Fig. 10 (g) Iris, (h) Heart, and (i) Magic, It is clear that LNSMO converges faster than DE and other algorithms converge prematurely and SMO follows the same characteristic of LNSMO but the high value of SWCD. From Fig. 10j of the HTRU2 dataset PSO, GWO and GA converge sharply with a higher value of SWCD while DE takes more iterations to compare to LNSMO. From Fig. 10k of the Haberman dataset PSO, GWO and GA converge prematurely with a higher value of SWCD while LNSMO converges faster than DE. Overall results show that LNSMO converges faster than DE and other algorithms stuck to premature convergence, While SMO follows the same characteristic of LNSMO but with a higher value of SWCD.

5.3.4 Time complexity

Table 3 shows the CPU time in seconds for different algorithms to complete 20 runs. It shows that traditional algorithm like DE, GA, PSO, and GWO takes lesser time compare to the proposed algorithm but they produced sub-optimum value in SWCD result. On the other side, SMO does not trap to local optimum but takes more time. The reason behind more time taken by SMO is that it executes six different phases to balance exploration and exploitation. Our proposed algorithm produced better results compare to SMO with the addition of a very small time cost.

5.3.5 Comparison with recently published algorithms

In this comparison following algorithms are selected from the literature. VDEO [7], which is derived from DE involves the calculation of the variance of each feature and optional crossover strategy. AMADE [8] is the hybridization of a memetic algorithm with an adaptive DE mutation operator. PSOPC [5], in which hybridization of PSO is done with crossover operator and polygamous selection. GLPSOK [9], which is the hybridization of PSO-K-means algorithm with Gaussian estimation and Levy flight. WOAC [10], in which recently developed whale optimization is proposed for the clustering problem. KMCLUST [11], In which centroid obtain by a k-means algorithm is feed to the MBCO. TEABC_elite [12], in which K-means and chaotic parameters are combined with the previously proposed EABC_elite algorithm.

Table 5 shows the comparison between LNSMO and the recently published algorithm for best, mean, and standard

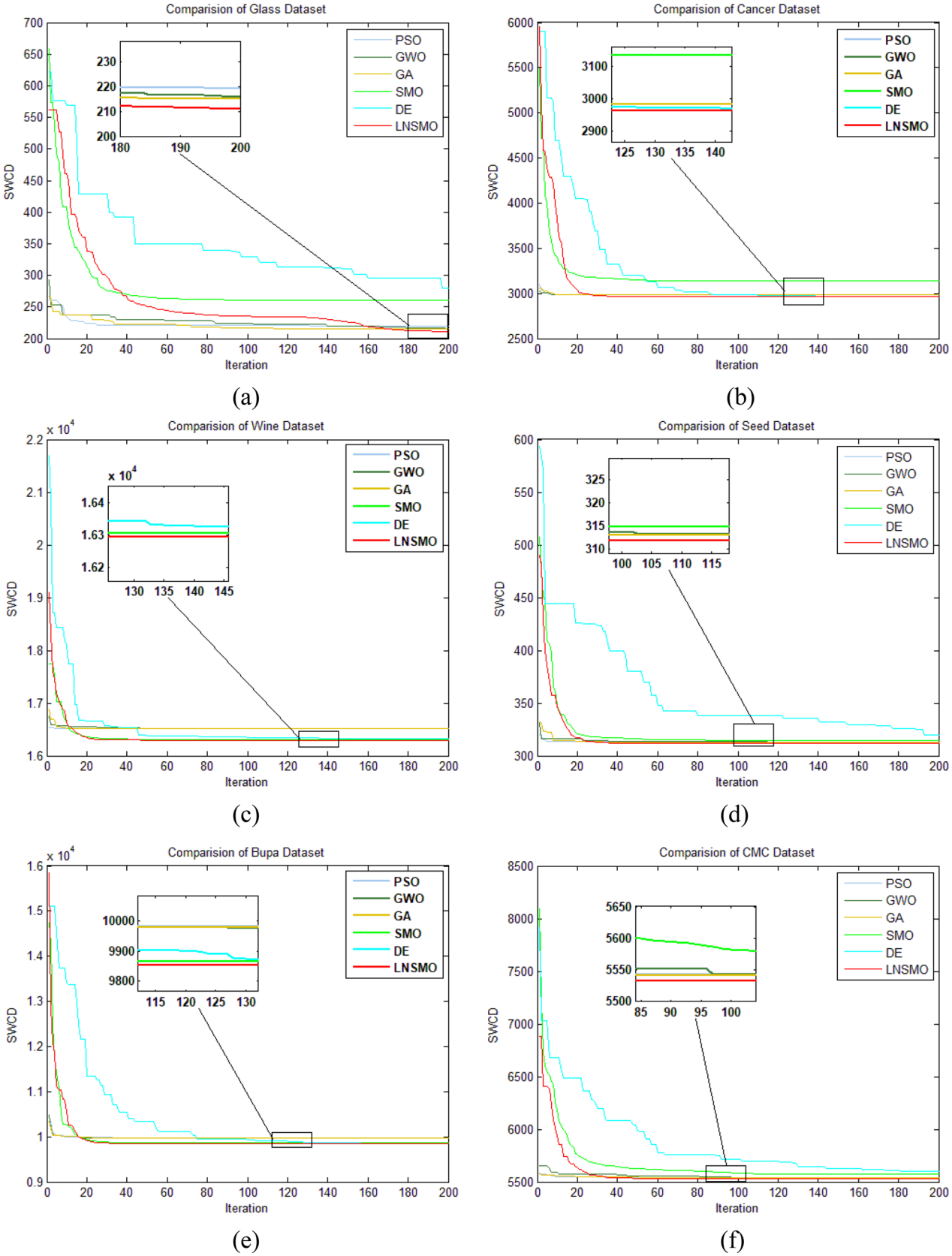

Fig. 10 Convergence plot for **a** Glass, **b** Cancer, **c** Wine, **d** Seed, **e** Bupa, **f** CMC, **g** Iris, **h** Heart, **i** Magic, **j** HTRU2, **k** Haberman

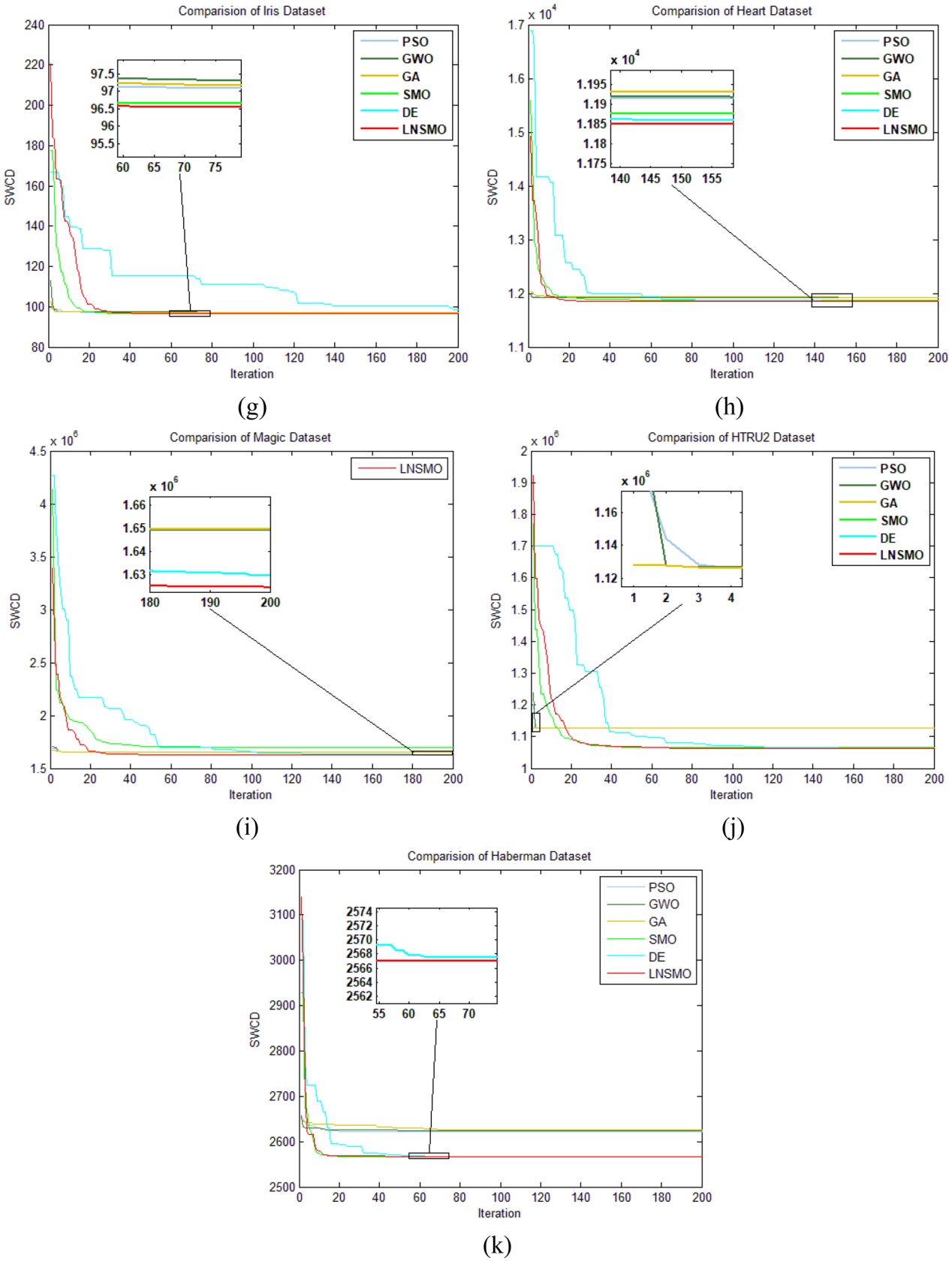

Fig. 10 (continued)

Table 5 Comparison with recently published algorithms over best, mean, and standard deviation of SWCD

	SWCD	GLASS	CANCER	Wine	Bupa	CMC	Iris	Haberman
VDEO [7]	Best	210.4	2964.41	16,292.43	–	–	96.54	**2566.99**
	Mean	213.62	2964.43	16,293.56	–	–	96.54	**2566.99**
	Std	1.99	0.02	0.87	–	–	**0**	**0**
AMADE [8]	Best	**210.17**	2964.393	16,292.28	–	5532.404	96.544	–
	Mean	211.214	2964.522	16,292.82	–	5532.62	96.549	–
	Std	**1.174**	0.091	**0.395**	–	0.423	0.004	–
PSOPC [5]	Best	210.433	**2964.387**	**16,292.18**	**9851.721**	**5532.185**	96.6555	**2566.989**
	Mean	219.28	**2964.387**	16,292.54	**9851.721**	5532.197	96.6555	**2566.989**
	Std	11.32426	**4.67E − 13**	0.60747	3.66E − 05	0.011096	2.92E − 14	9.33E − 13
GLPSOK [9]	Best	–	–	–	–	–	–	–
	Mean	215.14	–	16,295	–	5532.3	96.655	–
	Std	1.6923	–	8.9799	–	0.24391	3.74E-14	–
WOAC [10]	Best	–	–	–	–	–	–	–
	Mean	231.2912	3036.12	16,295.00	–	5539.72	96.7993	–
	Std	4.51	0.2	0.72	–	0.79	0.1	–
KMCLUST [11]	Best	215.23	2971.01	16,400.00	–	5,678.40	**95.19**	–
	Mean	221	2995.43	21,479.00	–	5,684.60	**95.98**	–
	Std	13.1	2.14	41.66	–	1.99	3.46	–
TEABC_elite [12]	Best	–	–	–	–	–	–	–
	Mean	**210.75**	2965.41	16,293.2	–	5534.2	96.65	–
	Std	3.14	9.76	3.27	–	1.81	0.0031	–
LNSMO	Best	210.464	**2964.387**	**16,292.18**	**9851.721**	**5532.185**	96.65548	**2566.989**
	Mean	212.768	**2964.387**	**16,292.44**	**9851.721**	**5532.185**	96.65548	**2566.989**
	Std	2.12	4.70E-09	0.466	**4.60E-09**	**7.64E-06**	4.36E-09	3.04E-09

deviation of SWCD results. In the table, bold values show better results compare to other algorithms. For the Glass dataset, AMADE produced better results in terms of the best and standard deviation of SWCD. TEABC_elite produced a better result in the mean value of SWCD. But our proposed algorithm produced comparable results in all quality measures with AMADE and TEABC_elite. For Cancer dataset proposed and the PSOPC algorithm produced similar results in best and mean values of SWCD with negligible standard deviation. In the Wine dataset, the proposed and the PSOPC algorithm produced similar results in the best value of SWCD while the proposed algorithm produced a better result in the mean value of SWCD. In Bupa dataset, the proposed and PSOPC produced similar results in the mean and best value of SWCD while both produced a negligible standard deviation of SWCD. In CMC dataset proposed and PSOPC algorithm produced a similar value of best SWCD. But proposed algorithm produced better results in the mean and standard deviation of SWCD compare to PSOPC and other algorithms. For the Iris dataset, KMCLUST produced better results of best and mean value of SWCD compare to all algorithms. But results produced by the proposed algorithm are nearer to other and KMCLUST algorithms. For the Haberman dataset, VDEO, PSOPC, and the proposed algorithm produced similar results with negligible standard deviation. From overall observation, it is concluded that the proposed algorithm produced better or similar results with previously published algorithms. In Table 5 bold text shows the best results compare to competent algorithm for SWCD measure.

Table 6 shows the comparison of LNSMO with a previously published algorithm for accuracy. The proposed algorithm produced better accuracy in Glass, Wine, Bupa, and CMC datasets for all the algorithms, and similar results with the Cancer and Iris dataset. For the Haberman dataset, the accuracy of the proposed algorithm is comparable with published results.In Table 6 bold text shows the best results compare to competent algorithm for accuracy measure.

5.4 Statistical significance analysis

For testing of the best algorithm, an unpaired t-test is performed based on mean SWCD between best and second-best

Table 6 Comparison with recently published algorithms based on Accuracy

Accuracy	GLASS	CANCER	Wine	Bupa	CMC	Iris	Haberman
VDEO [7]	0.5133	**0.9649**	0.7191			**0.9**	**0.5196**
AMADE [8]	0.6308	**0.96486**	0.719	–	0.4562	**0.9**	–
PSOPC [5]	–	**0.96486**	0.714	0.4956	0.3211	**0.9**	**0.5196**
GLPSOK [9]	0.5352	–	0.71685	–	0.39443	**0.9**	–
WOAC [10]	–	–	–	–	–	–	–
KMCLUST [11]	–	–	–	–	–	–	–
TEABC_elite [12]	–	–	–	–	–	–	–
LNSMO	**0.7182**	**0.9649**	**0.7211**	**0.5082**	**0.5537**	**0.9**	0.5088

algorithms. Confidence Interval (CI) between the two means is calculated based on data size equal to 20 and 95% of confidence level. In hypothesis testing, a two-tailed P-value of the t-test is the probability of finding extreme results when the null hypothesis for a given test is true. The smaller value of P supports an alternative hypothesis [42].

The confidence interval and P-value of each dataset are used to interpret a significant level of the proposed LNSMO algorithm with the second-best algorithm. The results are highly statically significant (HSS) when $P \leq 0.01$. If $P \leq 0.05$ statically significant (SS), and when $P > 0.10$ not statically significant (NSS).

Table 7 shows the result of the unpaired t-test between the best and second-best algorithm based on the mean of SWCD results. For all the dataset two-tailed P-value is less than 0.01 means, LNSMO is highly statically significant compared to all second-best algorithms except Bupa and Magic dataset. In Bupa and Magic dataset, P-value is less than 0.05 means, LNSMO is statically significant compared to the second-best algorithm DE.

6 Conclusion and future perspective

In this paper search process of the spider monkey optimization algorithm is improved with a local neighbour search method. To refined search space, the neighbour search is embedded with the local leader phase of SMO. Further global leader phase of the proposed algorithm is improved with a chaotic factor. The proposed LNSMO algorithm is applied to the clustering problem when the number of clusters known before. It is compared with the five traditional algorithms like PSO, GA, SMO, DE, and GWO, and tested on eleven data sets. The simulation results reveal that LNSMO outperforms with comparative algorithms in terms of SWCD, cluster quality measures, and convergence results for all the datasets. Statically unpaired t-test demonstrated that the proposed LNSMO algorithm is statistically significant. Further proposed LNSMO algorithm is compared with seven recently published hybrid meta-heuristic algorithms in which LNSMO produced better or similar results in SWCD and accuracy. Further LNSMO gives optimum results with a reasonable time cost.

Table 7 An unpaired t-test between best and second-best algorithm

Second Best	Data set	SE	t	CI	P (two-tailed)	Significance
PSO	Glass	1.211	−4.535	[−7.94293 − 3.04044]	<0.001	HSS
DE	Cancer	2.627	−6.534	[−22.4799 − 11.8446]	<0.0001	HSS
SMO	Wine	1.007	−11.247	[−13.3631 − 9.28643]	<0.0001	HSS
GWO	Seed	0.014	−77.095	[−1.08381 − 1.02835]	<0.0001	HSS
DE	Bupa	1.416	−3.443	[−7.74399 − 2.0094]	0.015	SS
GA	CMC	0.334	−31.169	[−11.0724 − 9.72184]	<0.0001	HSS
SMO	Iris	0.009	−49.898	[−0.454966 − 0.4194]	<0.0001	HSS
DE	Heart	0.505	−5.794	[−3.95056 − 1.90485]	<0.0001	HSS
DE	Magic	783.200	−3.477	[−4308.89 − 1137.62]	0.03	SS
DE	Htru2	80.810	−2.429	[−358.397 − 89.2541]	<0.01	HSS
DE	Haberman	0.002	−2.833	[−0.009659 − 0.0016]	<0.01	HSS

In the future proposed LNSMO algorithm can be extended with automatic data clustering when the number of the clusters is not known before. It is also extended with a multi-objective approach and parallel computing. In the future proposed algorithm, LNSMO can be applied to gene expression, image segmentation, etc.

Funding None.

Availability of data and material Not applicable.

Code availability Not applicable.

Declarations

Conflict of interest The authors declared that they have no conflict of interests.

References

1. Shabanzadeh P, Yusof R (2015) An efficient optimization method for solving unsupervised data classification problems. Comput Math Methods Med 2015:802754. https://doi.org/10.1155/2015/802754
2. Kao Y-T, Zahara E, Kao I-W (2008) A hybridized approach to data clustering. Expert Syst Appl 34:1754–1762. https://doi.org/10.1016/j.eswa.2007.01.028
3. Sloss AN, Gustafson S (2020) 2019 Evolutionary algorithms review. In: Banzhaf W, Goodman E, Sheneman L et al (eds) Genetic programming theory and practice XVII. Springer International Publishing, Cham, pp 307–344
4. Zhang Y, Agarwal P, Bhatnagar V et al (2013) Swarm intelligence and its applications. Sci World J 2013:528069. https://doi.org/10.1155/2013/528069
5. Sharma M, Chhabra JK (2019) An efficient hybrid PSO polygamous crossover based clustering algorithm. Evol Intell. https://doi.org/10.1007/s12065-019-00235-4
6. University of california irvine, ucirvine machine learning repository, http://archive.ics.uci.edu/ml/index.php
7. Alswaitti M, Albughdadi M, Isa NAM (2019) Variance-based differential evolution algorithm with an optional crossover for data clustering. Appl Soft Comput 80:1–17. https://doi.org/10.1016/j.asoc.2019.03.013
8. Mustafa HMJ, Ayob M, Nazri MZA, Kendall G (2019) An improved adaptive memetic differential evolution optimization algorithms for data clustering problems. PLoS ONE 14:1–28. https://doi.org/10.1371/journal.pone.0216906
9. Gao H, Li Y, Kabalyants P et al (2020) A novel hybrid PSO-K-means clustering algorithm using Gaussian estimation of distribution method and Lévy flight. IEEE Access 8:122848–122863. https://doi.org/10.1109/ACCESS.2020.3007498
10. Nasiri J, Khiyabani FM (2018) A whale optimization algorithm (WOA) approach for clustering. Cogent Math Stat 5:1483565. https://doi.org/10.1080/25742558.2018.1483565
11. Das P, Das DK, Dey S (2018) A modified bee colony optimization (MBCO) and its hybridization with k-means for an application to data clustering. Appl Soft Comput 70:590–603. https://doi.org/10.1016/j.asoc.2018.05.045
12. Du Z, Han D, Li K-C (2019) Improving the performance of feature selection and data clustering with novel global search and elite-guided artificial bee colony algorithm. J Supercomput 75:5189–5226. https://doi.org/10.1007/s11227-019-02786-w
13. Das S, Abraham A, Konar A (2008) Automatic clustering using an improved differential evolution algorithm. IEEE Trans Syst Man, Cybern - Part A Syst Humans 38:218–237
14. Bouyer A, Hatamlou A (2018) An efficient hybrid clustering method based on improved cuckoo optimization and modified particle swarm optimization algorithms. Appl Soft Comput 67:172–182. https://doi.org/10.1016/j.asoc.2018.03.011
15. Yang L, Zhang W, Lai Z, Cheng Z (2018) A particle swarm clustering algorithm based on tree structure and neighbourhood. In: Li K, Li W, Chen Z, Liu Y (eds) Computational intelligence and intelligent systems. Springer, Singapore, pp 67–85
16. Kushwaha N, Pant M (2019) A teaching–learning-based particle swarm optimization for data clustering. In: Tanveer M, Pachori RB (eds) Machine intelligence and signal analysis. Springer, Singapore, pp 223–233
17. Kumar Y, Singh PK (2018) Improved cat swarm optimization algorithm for solving global optimization problems and its application to clustering. Appl Intell 48:2681–2697. https://doi.org/10.1007/s10489-017-1096-8
18. Prakash J, Singh PK (2018) Hybrid Gbest-guided artificial bee colony for hard partitional clustering. Int J Syst Assur Eng Manag 9:911–928. https://doi.org/10.1007/s13198-017-0684-7
19. Aljarah I, Mafarja M, Heidari AA et al (2020) Clustering analysis using a novel locality-informed grey wolf-inspired clustering approach. Knowl Inf Syst 62:507–539. https://doi.org/10.1007/s10115-019-01358-x
20. Dhal KG, Das A, Ray S, Das S (2019) A clustering based classification approach based on modified cuckoo search algorithm. Pattern Recognit Image Anal 29:344–359. https://doi.org/10.1134/S1054661819030052
21. Tang Y, Wang N, Lin J, Liu X (2019) Using improved glowworm swarm optimization algorithm for clustering analysis. In: 2019 18th International symposium on distributed computing and applications for business engineering and science (DCABES). pp 190–194. https://doi.org/10.1109/DCABES48411.2019.00054
22. Li Y, Cai J, Yang H et al (2019) A novel algorithm for initial cluster center selection. IEEE Access 7:74683–74693. https://doi.org/10.1109/ACCESS.2019.2921320
23. Zabihi F, Nasiri B (2018) A novel history-driven artificial bee colony algorithm for data clustering. Appl Soft Comput 71:226–241. https://doi.org/10.1016/j.asoc.2018.06.013
24. Tripathi AK, Sharma K, Bala M (2018) A novel clustering method using enhanced grey wolf optimizer and MapReduce. Big Data Res 14:93–100. https://doi.org/10.1016/j.bdr.2018.05.002
25. Kumar V, Chhabra JK, Kumar D (2017) Grey wolf algorithm-based clustering technique. J Intell Syst 26:153–168. https://doi.org/10.1515/jisys-2014-0137
26. Hassanzadeh T, Meybodi MR (2012) A new hybrid approach for data clustering using firefly algorithm and K-means. In: The 16th CSI international symposium on artificial intelligence and signal processing (AISP 2012). pp 7–11. https://doi.org/10.1109/AISP.2012.6313708
27. Jadhav AN, Gomathi N (2018) WGC: Hybridization of exponential grey wolf optimizer with whale optimization for data clustering. Alex Eng J 57:1569–1584. https://doi.org/10.1016/j.aej.2017.04.013
28. Ghany KKA, AbdelAziz AM, Soliman THA, Sewisy AAE-M (2020) A hybrid modified step whale optimization algorithm with tabu search for data clustering. J King Saud Univ - Comput Inf Sci. https://doi.org/10.1016/j.jksuci.2020.01.015
29. Li Y, Ni Z, Jin F et al (2018) Research on clustering method of improved glowworm algorithm based on good-point set. Math Probl Eng 2018:8724084. https://doi.org/10.1155/2018/8724084

30. Isimeto R, Yinka-Banjo C, Uwadia CO, Alienyi DC (2017) An enhanced clustering analysis based on glowworm swarm optimization. In: 2017 IEEE 4th International conference on soft computing & machine intelligence (ISCMI). pp 42–49. https://doi.org/10.1109/ISCMI.2017.8279595
31. Neshat M, Yazdi SF, Yazdani D, Sargolzaei M (2012) A new cooperative algorithm based on PSO and K-means for data clustering. J Comput Sci 8:188–194. https://doi.org/10.3844/jcssp.2012.188.194
32. Saida IB, Nadjet K, Omar B (2014) A New algorithm for data clustering based on cuckoo search optimization. In: Pan J-S, Krömer P, Snášel V (eds) Genetic and evolutionary computing. Springer International Publishing, Cham, pp 55–64
33. Bansal JC, Sharma H, Jadon SS, Clerc M (2014) Spider monkey optimization algorithm for numerical optimization. Memetic Comput 6:31–47. https://doi.org/10.1007/s12293-013-0128-0
34. Tang L, Liu J (1999) A comparison of tabu search and local search methods for single machine scheduling with ready tim. IFAC Proc 32:6127–6132
35. Misagh Rahbari AJ A hybrid simulated annealing algorithm for travelling salesman problem with three neighbor generation structures. In: 10th International conference of iranian operations research society (ICIORS 2017), University of Mazandaran, Babolsar, Iran. https://hal.archives-ouvertes.fr/hal-01962049
36. Sharma A, Sharma A, Panigrahi BK et al (2016) Ageist spider monkey optimization algorithm. Swarm Evol Comput 28:58–77. https://doi.org/10.1016/j.swevo.2016.01.002
37. Arasomwan M, Adewumi A (2013) On adaptive chaotic inertia weights in Particle Swarm Optimization. In: IEEE Symposium on swarm intelligence (SIS). pp 72–79. https://doi.org/10.1109/SIS.2013.6615161
38. Sharma N, Kaur A, Sharma H et al (2019) Chaotic spider monkey optimization algorithm with enhanced learning. In: Bansal JC, Das KN, Nagar A et al (eds) Soft computing for problem solving. Springer, Singapore, pp 149–161
39. Kwedlo W (2011) A clustering method combining differential evolution with the K-means algorithm. Pattern Recogn Lett 32:1613–1621. https://doi.org/10.1016/j.patrec.2011.05.010
40. Cura T (2012) A particle swarm optimization approach to clustering. Expert Syst Appl 39:1582–1588
41. Maulik U, Bandyopadhyay S (2000) Genetic algorithm-based clustering technique. Pattern Recognit 33:1455–1465. https://doi.org/10.1016/S0031-3203(99)00137-5
42. Figueiredo D (2013) When is statistical significance not significant? Braz Polit Sci Rev. https://doi.org/10.1590/S1981-38212013000100002