A Hybrid Shuffled Frog-Simulated Annealing Algorithm for Clustering

Nitesh Sureja[1]*   **Avani Vasant[1]**   **Nandini Chaudhari[1]**

[1]*Department of Computer Science and Engineering, Babaria Institute of Technology, Gujarat, India*
* Corresponding author's Email: nmsureja@gmail.com

Abstract: Data clustering is a collection of data objects similar to one another within the same cluster and dissimilar to the objects in other clusters. The shuffled frog-leaping algorithm is a nature-inspired algorithm that mimics the natural biological evolution process of frogs. This algorithm also consists of elements like local search and exchanging information globally. This algorithm faces the problem of converging in local optima due to the limitations of the local search method used to explore search space. In this paper, a hybrid shuffled frog-leaping algorithm is introduced for clustering. The proposed algorithm uses a simulated annealing search method instead of a simple local search to improve the search behavior for selecting fitter solutions required in each iteration. Six benchmark datasets are used to validate the performance of the proposed algorithm. Quality measures used are purity, entropy, completeness score (CS), homogeneity score (HS), and FMeasure (FM). Fitness functions used to optimize are total within-cluster variance (TW) and the Silhouette coefficient (SC). Results obtained are compared with the results of twelve other state of the art algorithms. Results stored in the tables clearly shows that our proposed algorithm outperforms other algorithms in terms of quality. Results also prove that the proposed algorithm converges in the significantly less amount of time and eliminates local optima problem also.

Keywords: Nature inspired algorithms, Shuffled frog leaping algorithm, Data clustering, Simulated annealing.

1. Introduction

Clustering, also known as set partitioning, is a very widely used method in selecting locations, assignment and scheduling, partitioning of networks, routing and many more. It is also applied to the problems such as - computer graphics, image processing, pattern recognition and learning theory etc. We use clustering to group all objects into several mutually selected clusters to achieve the minimum or maximum objective function [1].

Various methods based on - Artificial intelligence, density, partitions are proposed and used for clustering. Hierarchical clustering is also one of the methods used. Centre-based clustering technique is also used in many data clustering algorithms. K-means algorithm is a perfect example of this. K-means algorithm is a very popular algorithm due to its fast convergence and simplicity in clustering of large datasets. The K-means algorithm has shortcomings such as initial state dependency, local optima.

In order to overcome local optima problems, several nature inspired algorithms such as, cuckoo search (CS) [2], firefly algorithm (FFA) [2], particle swarm optimization (PSO) [3], artificial bee colony (ABC) [4], grey wolf optimizer (GWO) [5], bat optimization (BAT) [2], multi-verse optimizer (MVO) [6], salp swarm algorithm (SSA) [7], ant colony optimization (ACO) [8], genetic algorithm [9] etc. have been used.

Eusuff et al presented the shuffled frog-leaping algorithm (SFLA) to solve the combinatorial and discrete optimization problems in 2006 [10]. This algorithm is based on the food searching (social) behavior of a population of frog. In their food searching process, frogs always try to get more food by using minimum efforts [11]. Frogs have a tendency to search for food in groups. This is the main idea employed in SFLA. The frogs form groups among themselves while searching for food. Each group is known as memeplex. The frogs in each

memeplex will have different culture. The frog, which is at the greatest distance from the food changes its place, based on the information it receives from the others frogs in its own memeplex and also from the other memplexes. Within each memeplex the frogs communicate among themselves to come with an idea which can contribute towards the global solution.

In most of the research based on the shuffled frog leaping algorithm (SFLA), only the movement of frogs towards food is only considered. A very little or no work done on a feature reproduction so far. No SFLA is proposed considering both features; food requirement and reproduction. We have tried to focus on reproduction feature also along with food requirement. We have used simulated annealing concept here to improve reproduction in frogs [12]. All frogs have defined body temperature required for the reproduction. All frogs have assigned a fitness value considering this factor. The body temperature of the frogs decides the fitness of the frogs required for reproduction. The frogs with less body temperature (less fit) will move towards food while others take part in the reproduction process. In this way we have tried to improve the effectiveness of the memetic evolution step and in turn the solutions produced.

In this research, for a given data set the hybrid SFLA is used to find the cluster centers. The hybrid SFLA algorithm uses this data set and the cluster centers are obtained. In order to evaluate the performance of the hybrid SFLA algorithm, entropy, purity, harmony score (HS), completeness score (CS) and FMeasure measures are used.

The paper is organized as the discussion of related work in section 2, basic introduction of SFLA, and proposed algorithm in section 3, experiments, results, and performance evaluation is presented in Section 4. We conclude the paper in Section 5 with the summary of observations.

2. Related work

Many nature inspired algorithms are developed by the researchers for clustering. In this section, some important recent contributions towards clustering are discussed.

X. Xiao and H. huang presents an improved ACO clustering routing algorithm for Underwater Wireless Sensor Networks in 2020 [8]. They used an improved of the heuristic information with a new evaporation parameter for the pheromone update to improve ant searching scope. They found that their algorithm gives good performance in reducing the energy consumption but st the cost of increased packet loss ratio.

Alhenaki and hosny presented a hybrid genetic-shuffled frog leaping algorithm for clustering text documents [9]. Genetic algorithm is used for features selection task, while a shuffled frog-leaping algorithm is used to handle the clustering task in this approach. A popular text document dataset "20Newsgroup" is used for evaluation. This algorithm improved the text document clustering task, compared to classical K-means clustering. But, this improvement comes at the expense of longer computational time.

J. Rahimipour, S. Samet, M. Eftekhari and W.A. Chang presented a feature selection approach to reduce attributes by selecting the most informative features of a dataset [13]. The proposed approach hybridized the Binary-SFLA with the fuzzy-rough dependency degree (FRDD) measure which is used for feature subset evaluation with a new search strategy. Twenty two datasets, including nine high dimensional and large ones with nine classifiers used check the performance. The approach outperforms other approaches in terms of the number of selected features at the expense of computational time and stability.

Liauw, Khairuzzaman and Syarifudin [14] presented a metaheuristic grouping method of whale optimization algorithm with modification to weight changes based on humpback food hunting behaviour of whales. This approach produces good quality solutions in reasonable amount of time but faces the problem while used for large datasets.

Lukasik, Kowalski, Charytanowicz and Kulczycki [15] proposed a grasshopper optimization algorithm (GOA) – to generate accurate data clustering. They used internal clustering validation measure of Calinski-Harabasz index to measure the quality of the solutions.

Agbaje, Ezugwu and Els [16] presented an automatic data clustering using Hybrid Firefly Particle Swarm Optimization algorithm to improve convergence speed while dealing with many variables data clustering problems. This algorithm succeeds to produce the results in good times but stuck in optima in terms of big datasets.

The approaches discussed here, enhances the clustering performance in one or more ways. It is still necessary to provide a simple, efficient, robust algorithm which uses very few parameters and produces globally acceptable optimal solution. Therefore, this work proposes a shuffled frog-leaping algorithm, which uses simulated annealing instead of normal search in memetic evolution stage of the

algorithm, and provides better cluster solutions with faster convergence.

3. Shuffled frog leaping algorithm (SFLA)

Shuffled frog leaping algorithm (SFLA) is a population based memetic algorithm. It is based on the food searching behaviour of frogs. The shuffled frog leaping algorithm produces a solution by transforming the solutions (frogs) into a memetic evolution. All the frogs here create a memetic vector, which works as hosts for memes. In shuffled frog leaping algorithm, the population which is made of a set of solutions (frogs) is divided into memeplexes (subsets). Each memeplex (subset) is considered a different subset as frogs' culture in each memeplex is different. Each memeplex performs a local search. All frogs (solutions) present in each memeplex hold some idea (information), which they share to produce or evolve a process of evolution. This evolution process runs for a defined number of times. This shuffling and local search continues till an acceptable solution is generated [10, 11]. This algorithm uses a fitness value to find the quality of the individual.

3.1 Clustering using SFLA

A hybrid shuffled frog leaping algorithm (SFLA) is proposed here to find an appropriate fixed number of clusters, K, in a space R^N for clustering a set of fixed number (n) unlabeled points. This algorithm is hybridized by introducing simulated annealing search in place of normal local search in the process of memetic evolution within each memeplex (step 5 of the algorithm). Euclidean distance between the points and their respective cluster center is considered as clustering metric here.

The steps of the proposed algorithm based on [10, 11, 12] are described in detail as below.

Global exploration: frog-leaping algorithm

1. **Initialization:** initialize the number of the memplexes (m) and the number of frogs (n) in each memeplex. Initialize the sample size (F) of the swamp. The sample size, F, is given by $F = mn$. Also initialize the parameters of the simulated annealing algorithm used in place of local search. The parameters include, temperature assigned to each frog, weights (w_{min} and w_{max}) and two constants c_1 and c_2.
2. **Initialization of Population of frogs:** A population of F virtual frogs $A1, A2, ..., A(F)$ in the space $\Omega \in \Re^{Kd}$ is created. Here, K represents the number of clusters and d represents the number of dimensions of points. A vector of

decision variables $A(i) = \left(A_i^1, A_i^2, ..., A_i^{kd}\right)$ represents a frog (i) that is a candidate solution containing K cluster centers. Here, movement of the frogs is evaluated based on the fitness (performance) value $f(i)$, which is based on a temperature value assigned to them and calculated using formula 1.

$$f(i) = temp(i) \times \alpha \qquad (1)$$

Where, temp indicates the temperature assigned to each frog and α is a constant value depends on the number of the memplexes.

3. **Ranking of frogs:** Based on the fitness (performance) value, Sort all frogs (F) in descending order, and also store them in an array X. Store the position P_X of the best frog in the whole population.
4. **Creating memeplexes by partitioning of frogs:** divide the array (X) into m memeplexes where each memeplex (Y) contains n frogs, such that,

$$
\begin{aligned}
Y^l &= \left[\, A(j)^i, f(j)^i \,\middle|\, A(j)^i \right. \\
&= A(l + m(j - 1), f(j)^i) \\
&= f(l + m(j - l)), j = 1, ..., n \,], \\
& \qquad\qquad l = 1, ..., m \qquad (2)
\end{aligned}
$$

e.g., memeplex 1 will get rank 1 for $m = 3$, memeplex 2 will get rank 2, memeplex 3 will get rank 3, memeplex 1 will get rank 4, and so on.

5. **Memetic evolutions within each memeplex:** Evolve each memeplex, Y^l, $l = 1,...,m$ using simulating annealing (SA) search. After partitioning frogs to m memeplexes, evolve each memeplex and each of them should iterate N times. The algorithm will return to the global exploration for shuffling after the evolution of each memeplex. *Local (SA) exploration: frog-leaping algorithm*
   Step 5-a. Initialize a counter im to count the number of memplexes with the initial value 0, which in turn to be compared with the total number of memplexes (m). Initialize a counter iN to count the number of evolutionary steps with the initial value 0, which in turn to be compared with the maximum allowed steps (N). Define the best and worst frogs inside a memeplex as P_b and P_w respectively. Define the global best frog as P_g. We apply the evolution process in every iteration to improve the fitness of the worst frog only.
   Step 5-b. Initialize $im=im+1$.
   Step 5-c. Initialize $iN=iN+1$.
   Step 5-d. Improve the worst frog's position.

The worst fit frog improves its position based on the cost and wait assigned to it. The worst fit frog is adjusted using equations 3 and 4.

$$w = (temp \times (w_{max} - w_{min})) \times i / w_{max} \quad (3)$$

$$\begin{aligned} new\ position\ P_w = \\ current\ position\ P_w \times w \\ + c_1 \times rand(P_b - P_w) \\ + c_2 \times rand(P_g - P_w) \end{aligned} \quad (4)$$

Where w represents the weight associated and c_1 and c_2 represents constants values used.

Step 5-e. Repeat the computations given in equation three and four with respect to the global best frog, if the procedure does not produces a best solution (frog).

Step 5-f. Generate a new solution randomly for replacing worst frog with another frog having any random fitness, if still there is no improvement in the solution.

Step 5-g. If $iN<N$, go to step 5-c.

Step 5-h. If $im<m$, go to step 5-b. Otherwise, return to the global search to shuffle memeplexes.

6. **Memeplex shuffling:** Collect the frogs of all memplexes after the defined number of memetic evolution steps. Sort all frogs (solutions) in a descending order based on their fitness.

7. **Convergence checking:** Stop the algorithm, if any of the termination criteria is satisfied. Otherwise, repeat algorithm from the step two.

Pseudo-code of the proposed clustering algorithm is given in the figure 1.

3.2 Fitness evaluation

We use Silhouette coefficient (*SC*) and Total within Cluster Variance (*TW*) fitness functions available in the literature for evaluation. They are discussed in brief as below.

Silhouette coefficient (*SC*) [18]:

$$SC = \frac{\sum_{k=1}^{|K|}((b-a)/max(a,b))}{N} \quad (5)$$

Where a is the average distance between a point and the other is points in the same predicted cluster and b is the average distance between a point and the other points in the next nearest cluster. By maximizing SC, we obtain better results. We normalize the value of SC between 0 and 1 and then use the reversed value of the normalized *SC* ($1 - norm\ (SC)$) in the fitness function.

Total within Cluster Variance (*TW*) [18]:

$$TW = \sum_{n=1}^{N}\sum_{f=1}^{N} p_{nf}^2 - \sum_{k=1}^{K}\frac{1}{|pk|}\sum_{f=1}^{F}\left(\sum p_{kf}\right)^2 \quad (6)$$

Where, F is the number of features, p_{nf} is feature f of the point n, p_{kf} is feature f of the point k, and $|pk|$ is the number of points in cluster k. By minimizing *TW*, we obtain better results.

4. Results and discussions

4.1 Data sets

SFLA is implemented in MATLAB 2012a and run on an Intel i7, 2.8HZ, 8GB RAM System. We have performed experiments on 6 data sets which are used in the literature. The first 5 data sets are obtained from the UCI database repository [17] and the last one is collected from KEEL. Data sets are described in Table I and discussed in brief as below.

Iris data set has three clusters, 150 points and four attributes. It consists of 3 varieties of flowers. Glass data set has six clusters, 214 points and 9 attributes. This dataset contains the types of glass motivated by criminological investigation. Seeds data set has 3 clusters, 210 points and 7 attributes. It contains 7 kernels belongs to 3 varieties of wheat each consists of 70 elements. Wine data set has 3 clusters, 178 points and 13 attributes. It is an outcome of a chemical analysis of three different types of wines derived from three different cultivars. Heart data set has 2 clusters, 270 points and 13 attributes. This dataset consists of 209 instances with 8 health measures with four types of chest pain. Appendicitis dataset has 2 clusters, 106 points and 7 attributes. It represents 7 medical measures derived from 106 patients having appendicitis of two different classes.

4.2 Evaluation measures

In our experiments, we use five evaluation measures. They are, entropy, purity, completeness score (CS), homogeneity score (HS), and FMeasure. They are also considered as external measures. Entropy measures how the various semantic classes are distributed within each cluster, and is calculated by the following equation [18]:

$$Entropy = \sum_{j=1}^{k}\frac{(|P_j|)}{n}E(P_j) \quad (7)$$

Table 1. Data set Properties

Dataset	Points	Attributes	Classes
glass	214	9	6
wine	178	13	3
iris	150	4	3
seeds	210	7	3
heart	270	13	2
appendicitis	106	7	2

Table 2. Experimental settings of SFLA

Parameters	Value
Population size	50
Number of Memeplexes (m)	5
Number of frogs per memeplex(n)	10
Maximum Iterations per memeplex	50
Maximum Iterations	10

Where, $E(P_j)$ is the individual entropy of a cluster. The purity of the clustering is defined as [18]:

$$Purity = \frac{1}{n} \sum_{j=1}^{k} max_i \left(|T_i \cap P_j| \right) \qquad (8)$$

Where P_j presents all points assigned to cluster j, k is the number of clusters, and T_i is the true assignments of points in cluster i.

Homogeneity score (HS) is defined as [18]:

$$HS = 1 - \frac{H\left(\frac{T}{P}\right)}{H(T)} \qquad (9)$$

Where $H(T)$ is the clusters Entropy and $H(T/P)$ is the clusters conditional Entropy.

Completeness score (CS) is defined as [18]:

$$CS = 1 - \frac{H\left(\frac{P}{T}\right)}{H(P)} \qquad (10)$$

Where $H(P)$ is the cluster Entropy and $H(P/T)$ is the clusters conditional Entropy.

The hybrid shuffled frog leaping algorithm is implemented, and the experimental results are obtained with six datasets. The results are summarized in the tables 3 to 14. The tables show the best, worst, average, and standard deviation of the implementation of the fitness functions and values of measures. The results are compared with the algorithms, PSO, BAT, GWO, GA, SSA, MVO, FFA

1. Begin
2. Create the population of P individuals (solutions) randomly;
3. Calculate the fitness of each individuals (i);
4. Sort the population in descending order according to their fitness;
5. Group the population into several (m) subgroups (memeplexes);
6. For each subgroup (memeplex) do
7. Find out the worst and best individuals;
8. Update the worst individual position using defined equations;
9. Repeat the process for defined n iterations;
10. Combine the newly generated subgroups (memeplexes);
11. Sort the population in descending order according to their fitness;
12. Check if termination = true;
13. End

Fig. 1 Pseudo-code for proposed SFLA

and ABC.

We have compared the results of our proposed approach with some recently proposed hybrid SFLAs for clustering. We have compared the results with the algorithm (FuzzySFLA) proposed by R. Javed in 2018, where they combined the fuzzy-rough feature selection and SFLA. Comparison with the hybrid SFLA and GA (SFLAGA) proposed by yang in 2018 is also given. Results are also compared with the hybrid iBPSO_SFLA algorithm introduced by S. Rajamohana. We also compare the results with the SFLA for Feature Selection and Fuzzy Classifier Design (FCSFLA) proposed by I. Hodashinsky in 2019. We have implemented and executed all the above algorithms.

Tables 3 and 9 presents the results obtained with the iris data set. From the tables, we have observed that the proposed SFLA, GWO and FFA produce best results for TW fitness function. We have also observed that proposed SFLA, PSO, and FFA produce the best results for SC fitness function. Value of FMeasure indicates that SFLA and FFA generate the best accuracy in the results.

Table 4 and 10 presents the results obtained with the glass data set. From the tables, we have observed that the proposed SFLA and MVO produce the best results for TW fitness function. We have also observed that proposed SFLA, MVO and SSA produce the best results for SC fitness function. Value of FMeasure indicates that SFLA and MVO generate the best accuracy in the results.
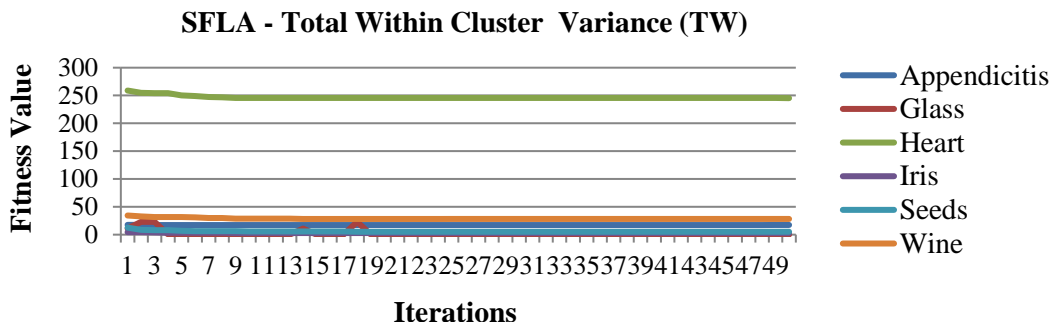
SFLA - Total Within Cluster Variance (TW)

Fig. 2 Result for proposed SFLA algorithm using TW Fitness Function
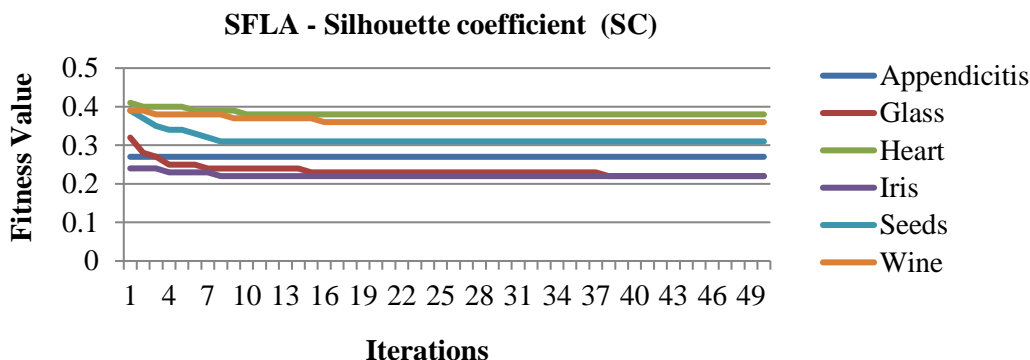
SFLA - Silhouette coefficient (SC)

Fig. 3 Result for proposed SFLA algorithm using SC Fitness Function

Table 3. Comparison of time, fitness and measures over iris data set

	Criteria	SFLA		PSO		GWO		GA		BAT		MVO		SSA	
		TW	SC	TW	SC	TW	SC	TW	SC	TW	SC	TW	SC	TW	SC
Fitness	Best	2.81	0.21	2.90	0.21	2.83	0.22	2.87	0.22	3.29	0.22	2.83	0.23	2.85	0.22
	Worst	4.49	0.24	3.83	0.25	3.82	0.24	4.72	0.23	3.91	0.22	3.24	0.25	4.51	0.24
	Avg.	2.79	0.22	2.96	0.22	2.87	0.21	3.04	0.22	3.39	0.22	2.89	0.21	2.98	0.22
	St. Dev.	0.14	0.12	0.15	0.09	0.10	0.10	0.10	0.09	0.13	0.09	0.11	0.11	0.17	0.13
Measure	Purity	0.97	0.88	0.95	0.86	0.96	0.86	0.95	0.86	0.91	0.86	0.96	0.86	0.96	0.83
	Entropy	0.14	0.31	0.15	0.33	0.14	0.32	0.14	0.33	0.19	0.34	0.14	0.33	0.15	0.37
	HS	0.88	0.77	0.85	0.67	0.86	0.86	0.76	0.77	0.80	0.76	0.86	0.77	0.85	0.63
	CS	0.87	0.92	0.85	0.91	0.86	0.86	0.86	0.89	0.82	0.88	0.86	0.89	0.85	0.89
	FM	0.92	0.91	0.86	0.86	0.86	0.86	0.86	0.81	0.81	0.80	0.81	0.82	0.85	0.83
Time	-	6.4	8.67	7.84	10.8	9.03	11.3	7.32	10.1	8.03	10.4	6.53	8.6	10.1	11.3

Table 4. Comparison of time, fitness and measures over glass data set

	Criteria	SFLA		PSO		GWO		GA		BAT		MVO		SSA	
		TW	SC	TW	SC	TW	SC	TW	SC	TW	SC	TW	SC	TW	SC
Fitness	Best	10.6	0.21	13.2	0.24	12.8	0.24	22.9	0.28	11.7	0.22	10.8	0.22	11.7	0.22
	Worst	14.6	0.32	16.6	0.29	16.7	0.29	26.7	0.31	15.6	0.27	15,7	0.33	15.9	0.33
	Avg.	12.4	0.23	14.9	0.26	13.8	0.29	23.5	0.28	13.1	0.23	12.5	0.23	12.7	0.00
	St. Dev.	0.09	0.13	0.09	0.09	0.13	0.11	0.09	0.13	0.08	0.11	0.08	0.11	0.43	0.09
Measure	Purity	0.61	0.65	0.53	0.47	0.41	0.40	0.46	0.44	0.51	0.48	0.51	0.48	0.7	0.44
	Entropy	0.51	0.63	0.59	0.66	0.76	0.76	0.67	0.68	0.60	0.63	0.59	0.64	0.17	0.71
	HS	0.35	0.26	0.29	0.22	0.10	0.09	0.20	0.19	0.29	0.25	0.29	0.24	0.28	0.16
	CS	0.78	0.80	0.35	0.45	0.78	0.80	0.44	0.42	0.40	0.50	0.35	0.5	0.1	0.51
	FM	0.96	0.94	0.92	0.90	0.91	0.91	0.92	0.93	0.91	0.92	0.95	0.93	0.96	0.95
Time	-	8.40	8.6	11.6	10.8	10.5	11.7	11.1	13.8	11.3	14.0	8.35	10.9	11.3	8.81

Table 5. Comparison of time, fitness and measures over heart data set

	Criteria	SFLA		PSO		GWO		GA		BAT		MVO		SSA	
		TW	SC	TW	SC	TW	SC	TW	SC	TW	SC	TW	SC	TW	SC
Fitness	Best	244	0.29	245	0.29	248	0.39	245	0.39	246	0.39	247	0.39	248	0.39
	Worst	255	0.40	257	0.41	254	0.39	258	0.41	255	0.41	260	0.41	261	0.41
	Avg.	248	0.39	248	0.39	249	0.39	246	0.38	246	0.38	248	0.39	243	0.38
	St. Dev.	0.20	0.23	0.17	0.19	0.21	0.20	0.17	0.24	0.21	0.20	0.16	0.18	0.16	0.14
Measure	Purity	0.95	0.93	0.90	0.86	0.85	0.92	0.93	0.86	0.93	0.90	0.92	0.92	0.90	0.90
	Entropy	0.59	0.60	0.67	0.71	0.70	0.63	0.63	0.71	0.63	0.66	0.64	0.63	0.61	0.61
	HS	0.19	0.18	0.09	0.13	0.14	0.07	0.06	0.13	0.06	0.09	0.08	0.07	0.06	0.04
	CS	0.17	0.16	0.10	0.14	0.15	0.11	0.06	0.14	0.07	0.13	0.08	0.11	0.06	0.09
	FM	0.96	0.95	0.91	0.87	0.86	0.93	0.91	0.89	0.92	0.91	0.93	0.91	0.91	0.89
Time	-	15.1	20.8	15.1	21.4	15.0	20.1	15.2	20.6	15.2	20.8	13.4	15.7	12.4	16.2

Table 6. Comparison of time, fitness and measures over seeds data set

	Criteria	SFLA		PSO		GWO		GA		BAT		MVO		SSA	
		TW	SC	TW	SC	TW	SC	TW	SC	TW	SC	TW	SC	TW	SC
Fitness	Best	5.17	0.28	5.42	0.29	6.52	0.32	5.35	0.33	5.22	0.30	5.20	0.31	5.35	0.30
	Worst	9.81	0.38	9.71	0.37	7.19	0.33	12.6	0.39	8.41	0.42	11.63	0.42	11.7	0.39
	Avg.	5.44	0.30	6.20	0.33	6.63	0.32	5.83	0.32	5.72	0.31	5.98	0.32	5.51	0.29
	St. Dev.	0.08	0.10	0.11	0.13	0.12	0.10	0.11	0.12	0.11	0.13	0.11	0.14	0.12	0.12
Measure	Purity	0.89	0.88	0.87	0.73	0.73	0.74	0.87	0.63	0.88	0.80	0.88	0.8	0.86	0.87
	Entropy	0.32	0.35	0.37	0.53	0.53	0.50	0.36	0.62	0.36	0.43	0.35	0.43	0.38	0.36
	HS	0.68	0.66	0.63	0.47	0.47	0.50	0.65	0.38	0.64	0.57	0.64	0.57	0.62	0.64
	CS	0.69	0.66	0.63	0.54	0.51	0.61	0.65	0.50	0.64	0.64	0.64	0.64	0.62	0.64
	FM	0.97	0.96	0.93	0.94	0.93	0.94	0.95	0.91	0.93	0.91	0.95	0.95	0.91	0.90
Time	-	10.2	10.1	10.1	11.0	10.5	12.3	11.5	14.8	10.5	14.3	8.99	11.1	8.54	11.8

Table 7. Comparison of time, fitness and measures over wine data set

	Criteria	SFLA		PSO		GWO		GA		BAT		MVO		SSA	
		TW	SC	TW	SC	TW	SC	TW	SC	TW	SC	TW	SC	TW	SC
Fitness	Best	27.3	0.34	28.5	0.37	31.7	0.38	28.2	0.36	27.5	0.36	27.4	0.35	28.0	0.35
	Worst	34.6	0.38	36.2	0.40	32.4	0.39	34.3	0.39	32.3	0.39	32.8	0.4	33.4	0.39
	Avg.	28.1	0.35	29.8	0.36	31.9	0.38	28.8	0.37	27.9	0.36	28.4	0.36	27.9	0.35
	St. Dev.	0.13	0.12	0.12	0.12	0.12	0.15	0.12	0.11	0.12	0.14	0.11	0.13	0.12	0.12
Measure	Purity	0.93	0.76	0.88	0.61	0.71	0.60	0.90	0.61	0.92	0.72	0.92	0.61	0.89	0.62
	Entropy	0.25	0.53	0.32	0.59	0.51	0.61	0.29	0.58	0.23	0.46	0.24	0.57	0.3	0.56
	HS	0.77	0.59	0.67	0.41	0.49	0.49	0.70	0.42	0.77	0.53	0.76	0.42	0.70	0.44
	CS	0.75	0.63	0.67	0.63	0.51	0.61	0.70	0.62	0.76	0.71	0.76	0.67	0.7	0.68
	FM	0.97	0.96	0.91	0.87	0.86	0.93	0.91	0.89	0.95	0.91	0.95	0.95	0.91	0.95
Time	-	34.6	0.38	36.5	0.40	32.5	0.39	34.4	0.39	32.4	0.39	7.56	11.0	7.23	10.5

Table 8. Comparison of time, fitness and measures over appendicitis data set

	Criteria	SFLA		PSO		GWO		GA		BAT		MVO		SSA	
		TW	SC	TW	SC	TW	SC	TW	SC	TW	SC	TW	SC	TW	SC
Fitness	Best	17.4	0.27	17.4	0.28	17.5	0.28	17.4	0.29	17.4	0.28	17.4	0.27	17.4	0.27
	Worst	18.1	0.28	18.0	0.28	17.7	0.28	17.5	0.27	18.3	0.28	18.1	0.28	18.3	0.29
	Avg.	17.4	0.27	17.4	0.27	17.5	0.28	17.4	0.27	17.4	0.27	17.4	0.27	17.4	0.26
	St. Dev.	0.19	0.18	0.14	0.20	0.14	0.20	0.17	0.17	0.16	0.20	0.17	0.16	0.17	0.18
Measure	Purity	0.87	0.89	0.81	0.80	0.81	0.81	0.82	0.84	0.80	0.82	0.80	0.80	0.8	0.82
	Entropy	0.55	0.65	0.57	0.71	0.57	0.67	0.55	0.67	0.57	0.67	0.57	0.65	0.57	0.66
	HS	0.21	0.01	0.21	0.01	0.21	0.06	0.23	0.13	0.21	0.08	0.21	0.01	0.21	0.07
	CS	0.17	0.04	0.18	0.04	0.17	0.09	0.20	0.19	0.17	0.11	0.17	0.04	0.17	0.12
	FM	0.96	0.96	0.95	0.91	0.92	09.0	0.95	0.91	0.91	0.90	0.93	0.95	0.93	0.95
Time	-	5.41	6.43	5.41	9.60	5.51	7.00	6.85	10.4	5.70	9.55	6.27	6.46	6.15	6.38

Table 9. Comparison of time, fitness and measures over iris data set

	Criteria	FuzzySFLA		iBPSO_SFLA		SFLAGA		FCSFLA		FFA		ABC	
		TW	SC	TW	SC	TW	SC	TW	SC	TW	SC	TW	SC
Fitness	Best	3.04	0.22	3.01	0.23	2.84	0.24	2.83	0.25	2.83	0.21	3.29	0.22
	Worst	5.69	0.23	4.15	0.24	3.63	0.26	4.98	0.24	3.8	0.22	3.91	0.22
	Average	3.24	0.22	3.07	0.21	2.97	0.21	2.94	0.21	2.93	0.21	3.39	0.22
	St. Dev.	0.14	0.16	0.19	0.1	0.13	0.12	0.11	0.15	0.15	0.12	0.13	0.09
Measure	Purity	0.96	0.86	0.93	0.86	0.96	0.86	0.96	0.86	0.96	0.86	0.91	0.86
	Entropy	0.12	0.24	0.19	0.14	0.14	0.13	0.14	0.14	0.14	0.23	0.19	0.24
	HS	0.88	0.76	0.81	0.86	0.86	0.87	0.86	0.86	0.86	0.77	0.80	0.76
	CS	0.88	0.88	0.82	0.86	0.86	0.87	0.86	0.86	0.86	0.88	0.82	0.88
	FM	0.73	0.75	0.80	0.81	0.78	0.80	0.86	0.84	0.94	0.89	0.81	0.80
Time	-	10.1	11.9	10.7	11.5	15.2	19.8	8.09	10.3	7.71	10.4	8.03	10.41

Table 10. Comparison of time, fitness and measures over glass data set

	Criteria	FuzzySFLA		iBPSO_SFLA		SFLAGA		FCSFLA		FFA		ABC	
		TW	SC	TW	SC	TW	SC	TW	SC	TW	SC	TW	SC
Fitness	Best	13.3	0.29	12.1	0.32	13.1	0.24	11.0	0.22	12.8	0.23	13.2	0.24
	Worst	15.7	0.39	15.8	0.41	19.3	0.26	15.3	0.3	17.7	0.27	15.6	0.28
	Average	14.4	0.27	14.3	0.33	15.0	0.24	12.9	0.23	14.9	0.26	14.9	0.26
	St. Dev.	0.1	0.1	0.08	0.1	0.09	0.1	0.06	0.13	0.07	0.12	0.09	0.09
Measure	Purity	0.43	0.41	0.46	0.42	0.51	0.48	0.54	0.49	0.52	0.47	0.53	0.47
	Entropy	0.73	0.75	0.67	0.72	0.59	0.64	0.57	0.63	0.6	0.66	0.59	0.66
	HS	0.35	0.15	0.21	0.14	0.3	0.24	0.33	0.26	0.28	0.21	0.29	0.22
	CS	0.35	0.33	0.4	0.33	0.35	0.46	0.38	0.51	0.36	0.45	0.35	0.45
	FM	0.92	0.91	0.89	0.89	0.91	0.93	0.92	0.90	0.91	0.91	0.89	0.90
Time	-	8.63	10.2	8.43	8.81	13.7	16.0	8.3	8.49	13.8	16.2	10.6	10.7

Tables 5 and 11 presents the results obtained with the heart data set. From the table, we have observed that the proposed SFLA, iBPSO_SFLA and SFLAGA produce the best results for TW fitness function. We have also observed that proposed SFLA, PSO and iBPSO_SFLA produce the best results for SC fitness function. Value of FMeasure indicates that SFLA and iBPSO_SFLA generate the best accuracy in the results.

Tables 6 and 12 presents the results obtained with the seeds data set. From the table, the proposed SFLA, FuzzySFLA and MVO produce the best results for TW fitness function. We have also observed that proposed SFLA, PSO and FCSFLA produce the best results for SC fitness function. Value of FMeasure indicates that SFLA and iBPSO_SFLA generate the best accuracy in the results.

Table 7 and 13 presents the results obtained with the wine data set. From the table, we have observed that the proposed SFLA, BAT, MVO and FCSFLA produce the best results for TW fitness function. We have also observed that proposed SFLA, MVO, SSA and FFA produce the best results for SC fitness function. Value of FMeasure indicates that SFLA and MVO generate the best accuracy in the results.

Table 8 and 14 presents the results obtained with the appendicitis data set. From the table, we have observed that the proposed SFLA, PSO, GA and FFA produce the best results for TW. We have also observed that proposed SFLA, MVO, SSA and iBPSO_SFLA algorithms produce best results for SC fitness function. Value of FMeasure indicates that SFLA and MVO generate the best accuracy in the results.

Figure 2 and 3 present graphical representation of the results obtained with hybrid SFLA using TW and SC fitness functions.

5. Conclusion

The paper proposes a hybrid shuffled frog leaping algorithm for the partitional clustering. This algorithm uses simulated annealing algorithm in place of normal local search in the memetic evolution step. Besides the description of the method, the results of its experimental evaluation were also discussed. It was established that hybrid SFLA gives very good performance with respect to other algorithms, in terms of quality, stability and time. The impact of important SFLA algorithm's parameters is also examined. Further studies within this paper's scope should include a more detailed analysis of the effects of population size and other coefficients used.

Conflicts of interest

The authors declare no conflict of interest.

Table 11. Comparison of time, fitness and measures over heart data set for hybrid SFLA algorithms

	Criteria	FuzzySFLA TW	SC	iBPSO_SFLA TW	SC	SFLAGA TW	SC	FCSFLA TW	SC	FFA TW	SC	ABC TW	SC
Fitness	Best	247	0.39	245	0.29	245	0.31	247	0.39	246	0.38	248	0.39
	Worst	259	0.4	258	0.41	258	0.4	260	0.41	254	0.39	255	0.40
	Average	248	0.39	245	0.39	246	0.39	247	0.39	246	0.38	248	0.39
	St. Dev.	0.2	0.19	0.28	0.3	0.19	0.2	0.22	0.24	0.19	0.2	0.20	0.23
Measure	Purity	0.88	0.94	0.86	0.94	0.86	0.86	0.92	0.86	0.86	0.9	0.90	0.90
	Entropy	0.68	0.62	0.51	0.53	0.51	0.57	0.64	0.71	0.71	0.66	0.65	0.65
	HS	0.11	0.05	0.13	0.06	0.13	0.13	0.08	0.13	0.13	0.09	0.09	0.09
	CS	0.12	0.06	0.14	0.06	0.14	0.14	0.08	0.14	0.14	0.13	0.09	0.09
	FM	0.87	0.85	0.95	0.94	0.94	0.93	0.89	0.85	0.86	0.85	0.89	0.89
Time	-	15.0	20.2	15.1	20.1	26.0	40.5	15.1	20.2	19.4	30.8	15.1	20.7

Table 12. Comparison of time, fitness and measures over seeds data set

	Criteria	FuzzySFLA TW	SC	iBPSO_SFLA TW	SC	SFLAGA TW	SC	FCSFLA TW	SC	FFA TW	SC	ABC TW	SC
Fitness	Best	5.20	0.3	5.95	0.31	5.61	0.3	5.29	0.29	5.34	0.3	6.52	0.32
	Worst	10.4	0.37	11.4	0.34	8.38	0.36	9.44	0.39	7.28	0.38	7.19	0.33
	Average	5.55	0.31	6.15	0.31	5.90	0.31	5.72	0.29	5.72	0.31	6.63	0.32
	St. Dev.	0.11	0.14	0.12	0.12	0.11	0.09	0.12	0.1	0.13	0.08	0.12	0.10
Measure	Purity	0.89	0.88	0.85	0.82	0.85	0.84	0.88	0.85	0.86	0.87	0.73	0.74
	Entropy	0.34	0.39	0.37	0.45	0.39	0.44	0.34	0.43	0.38	0.44	0.53	0.50
	HS	0.58	0.4	0.63	0.35	0.6	0.66	0.66	0.67	0.61	0.66	0.47	0.50
	CS	0.58	0.45	0.64	0.46	0.61	0.67	0.66	0.67	0.62	0.67	0.51	0.61
	FM	0.96	0.93	0.83	0.80	0.85	0.81	0.82	0.84	0.84	0.85	0.89	0.85
Time	-	11.4	14.3	17.3	15.3	20.3	24.8	10.2	13.7	15.4	18.6	10.4	12.3

Table 13. Comparison of time, fitness and measures over wine data set

	Criteria	FuzzySFLA TW	SC	iBPSO_SFLA TW	SC	SFLAGA TW	SC	FCSFLA TW	SC	FFA TW	SC	ABC TW	SC
Fitness	Best	29.2	0.36	28.6	0.39	28.1	0.37	27.5	0.37	28.5	0.36	31.7	0.38
	Worst	35.6	0.39	35.3	0.39	33.2	0.39	33.7	0.4	34.6	0.38	32.4	0.39
	Average	29.9	0.35	29.3	0.36	29.4	0.36	28.2	0.35	30.0	0.36	31.9	0.38
	St. Dev.	0.1	0.12	0.11	0.15	0.12	0.13	0.17	0.14	0.11	0.13	0.12	0.15
Measure	Purity	0.84	0.61	0.87	0.62	0.89	0.61	0.9	0.72	0.87	0.62	0.71	0.60
	Entropy	0.38	0.57	0.33	0.56	0.3	0.59	0.25	0.45	0.33	0.57	0.51	0.61
	HS	0.62	0.42	0.67	0.44	0.7	0.4	0.74	0.55	0.66	0.43	0.49	0.39
	CS	0.63	0.66	0.67	0.69	0.7	0.63	0.75	0.69	0.66	0.69	0.51	0.61
	FM	0.84	0.82	0.83	0.94	0.79	0.81	0.95	0.78	0.91	0.93	0.90	0.91
Time	-	10.2	19.9	10.1	12.6	18.4	24.9	10.1	14.1	14.5	19.1	10.3	10.7

Table 14. Comparison of time, fitness and measures over appendicitis data set

	Criteria	FuzzySFLA TW	SC	iBPSO_SFLA TW	SC	SFLAGA TW	SC	FCSFLA TW	SC	FFA TW	SC	ABC TW	SC
Fitness	Best	17.4	0.28	17.4	0.27	17.4	0.29	17.4	0.28	17.4	0.28	17.8	0.28
	Worst	17.7	0.28	17.9	0.28	17.7	0.28	17.7	0.28	17.6	0.28	18.1	0.28
	Average	17.4	0.27	17.5	0.27	17.4	0.27	17.4	0.27	17.4	0.27	17.4	0.27
	St. Dev.	0.15	0.14	0.16	0.26	0.16	0.21	0.16	0.18	0.13	0.19	0.19	0.18
Measure	Purity	0.81	0.85	0.82	0.92	0.88	0.88	0.88	0.82	0.81	0.86	0.80	0.80
	Entropy	0.56	0.6	0.55	0.66	0.57	0.71	0.57	0.66	0.56	0.71	0.57	0.71
	HS	0.23	0.17	0.23	0.08	0.21	0.01	0.21	0.08	0.22	0.01	0.21	0.01
	CS	0.19	0.21	0.21	0.21	0.17	0.04	0.17	0.1	0.18	0.04	0.17	0.04
	FM	0.86	0.85	0.83	0.86	0.81	0.84	0.85	0.86	0.95	0.91	0.81	0.83
Time	-	6.45	10.6	5.52	10.2	10.6	15.8	5.5	9.83	8.61	12.8	5.64	9.43

Author contributions

Conceptualization, NMS and ARV; methodology, NMS; software, NMS and NMC; validation, NMC and ARV; formal analysis, NMS; investigation, NMS; resources, ARV; data collection, NMS and ARV; writing—original draft preparation, NMS and ARV; writing—review and editing, NMS, ARV and NMC; visualization, NMS; supervision, NMS.

Acknowledgments

References

[1] A. K. Jain and M. N. Murty, "Data Clustering: A Review", ACM Computing Surveys, Vol. 31, No. 3, pp. 264–323, 1999.

[2] X. S Yang, "Nature-Inspired Metaheuristic Algorithms", Luniver Press, 2008.

[3] J. Kennady and R. C. Eberhart, "particle swarm optimization", In: IEEE Intl. conf. on neural networks, Vol. 4, pp. 1942–1948, 1995.

[4] Karaboga and A. Basturk, "On the performance of Artificial Bee Colony algorithm", Applied soft Computing, Vol. 8, No. 1, pp. 687–697, 2008.

[5] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer", Advances in Engineering software, pp. 198–226, 2014.

[6] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-Verse Optimizer: a nature inspired algorithm for global optimization", Neural Comput. Appl. Vol. 27, No. 2, pp. 495‑513, 2015.

[7] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems", Adv. Eng. Softw, Vol. 114, pp. 163‑191, 2017.

[8] X. Xiao and H. Huang, "A Clustering Routing Algorithm Based on Improved Ant Colony Optimization Algorithms for Underwater Wireless Sensor Networks", Algorithms, Vol. 13, No. 10, p. 250, 2006.

[9] L. Alhenaki and M. Hosny, "A Genetic-Frog Leaping Algorithm for Large Dataset Document Clustering", In: IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, pp. 1-4, 2019.

[10] M. Eusuff, K. Lansey, and F. Pasha, "Shuffled Frog Leaping Algorithm: A memetic meta-heuristic for discrete optimization", Engineering Optimization, Vol. 38, No. 2006, pp. 129-154, 2006.

[11] M. Karakorum and A. Babalik, "Data clustering with shuffled leaping frog algorithm (SFLA) for classification", In: International Conference on Intelligent Computing, Electronics Systems and Information Technology, pp. 25-26, 2015.

[12] S. Rathi, V. S. Rajput, and S. Gupta, "A hybrid evolution based optimization algorithm originated from the concept of SFLA and simulated annealing", In: 11th International Conference on Industrial and Information Systems (ICIIS), pp. 193-198, 2016.

[13] J. Rahimipour, S. Samet, M. Eftekhari, and W. A. Chang, "A Fuzzy-Rough Feature Selection based on Binary Shuffled Frog Leaping Algorithm", International Journal of Computer, Electrical, Automation, Control and Information Engineering, Vol. 12, No. 9, pp. 722 - 729, 2018.

[14] M. Liauw, Q. Khairuzzaman, and G. Syarifudin, "Whale Optimization Algorithm for Data Clustering", In: 7th International Conference on Cyber and IT Service Management (CITSM), Jakarta, Indonesia, pp. 1-6, 2019.

[15] S. Łukasik, P. Kowalski, M. Charytanowicz, and P. Kulczycki, "Data Clustering with Grasshopper Optimization Algorithm", In: Federated Conference on Computer Science and Information Systems, 2017.

[16] M. B. Agbaje, A. E Ezugwu, and R. Els, "Automatic data clustering using hybrid firefly particle swarm optimization algorithm", In: IEEE Access, Vol. 7, pp. 963-984, 2019.

[17] "UCI machine learning repository", http://archieve.ics.uci.edu/ml/.

[18] R. Qaddoura, H. Faris, I. Aljarah and P. A. Castillo, "EvoCluster: An Open-Source Nature-Inspired optimization Clustering Framework in Python", In: Applications of Evolutionary Computation. EvoApplications, Springer, Cham, Vol. 12104, pp. 20-36, 2020.