This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2937021, IEEE Access

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

An Enhanced Version of Black Hole Algorithm Via Levy Flight for Optimization and Data Clustering Problems

Haneen A. Abdulwahab[1], A. Noraziah[1, 2], AbdulRahman A. Alsewari[1], Sinan Q. Salih[3,4]

[1]Faculty of Computer Systems and Software Engineering, University Malaysia Pahang, 26300, Gambang, Pahang, Malaysia.
[2]IBM Center of Excellence, Universiti Malaysia Pahang, 26300, Gambang, Pahang, Malaysia.
[4]Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam
[3]Computer Science Department, College of Computer Science and Information Technology, University of Anbar, Ramadi, Iraq.

Corresponding author: Sinan Q. Salih (sinanq.salih@duytan.edu.vn).

ABSTRACT The processes of retrieving useful information from a dataset are an important data mining technique that is commonly applied, known as Data Clustering. Recently, nature-inspired algorithms have been proposed and utilized for solving the optimization problems in general, and data clustering problem in particular. Black Hole (BH) optimization algorithm has been underlined as a solution for data clustering problems, in which it is a population-based metaheuristic that emulates the phenomenon of the black holes in the universe. In this instance, every solution in motion within the search space represents an individual star. The original BH has shown a superior performance when applied on a benchmark dataset, but it lacks exploration capabilities in some datasets. Addressing the exploration issue, this paper introduces the levy flight into BH algorithm to result in a novel data clustering method "Levy Flight Black Hole (LBH)", which was then presented accordingly. In LBH, the movement of each star depends mainly on the step size generated by the Levy distribution. Therefore, the star explores an area far from the current black hole when the value step size is big, and vice versa. The performance of LBH in terms of finding the best solutions, prevent getting stuck in local optimum, and the convergence rate has been evaluated based on several unimodal and multimodal numerical optimization problems. Additionally, LBH is then tested using six real datasets available from UCI machine learning laboratory. The experimental outcomes obtained indicated the designed algorithm's suitability for data clustering, displaying effectiveness and robustness.

INDEX TERMS Optimization, Data Clustering, Black Hole, Levy Flight, Metaheuristic, Computational Intelligence.

I. INTRODUCTION

Data clustering is a method that consists of placing similar objects together, where like items are placed in one and different items are grouped in different ones. It is an unsupervised learning technique characterized by the grouping of objects in unspecified predetermined clusters. The conceptualization contrasts with classification, which is a form of supervised learning that involves objects being allocated to predetermined classes (clusters) [1]. Data clustering is widely used in many areas including data mining, statistical data analysis, machine learning, pattern recognition, image analysis, information retrieval, and more. This is due to clustering methods that can be categorized into various methods, such as partitional, hierarchical, density-based, grid-based, and model-based methods, accordingly[2].

Per the above methods, partitional clustering methods are the type that is commonly used, in which the K-means algorithm is an example of partitional and center-based clustering algorithms. Due to cluster centers being initialized, the k-means clustering algorithm is limited to the local optima[3]. Regardless, the past few decades have witnessed the development of many nature-inspired evolutionary algorithms in order to resolve engineering design optimization problems. They are known to emulate the behaviors of living things within nature, rendering them to be also described as Swarm Intelligence (SI) algorithms. SI algorithms typically search for global optima while being associated with speedy convergence[4].

Meanwhile, metaheuristic searching optimization is recently heavily discussed on in literature over wide-ranging engineering applications, such as power optimization control[5], robotic[6], communications and networking[7], engineering[8-12], information security[13, 14], and machine learning[15, 16]. Even though the approaches of the knowledge branch are characterizable by different concepts and inspirations, one fundamental attribute underlines their goal. All of the approaches make use of a selective searching process that is inspired by heuristic knowledge in the solution space to obtain a solution. The solution should optimize a given objective function or a set of objective functions in case of multi-optimization, provided that the set of constraints is maintained. These algorithms are highly attractive to researchers nowadays due to the fast enhancement of hardware speed and improved feasibility in solving many engineering problems. This is done by adhering to the heuristic searching conceptualization, with a simple design of objective function and constraints.

Various natural phenomena have led to the formulation of natural-inspired searching optimization algorithms[17, 18] such as hunting behavior of grey wolves[19]; krill herds[20]; black holes[21]; egg-laying behavior of cuckoos[22]; hunting behavior of bats[23]; food-searching behavior of bees[24]; and improvisation process of jazz musicians[25].

Recently, a meta-heuristic optimization called a “black hole” (BH) that mimics the black hole behavior of pulling in surrounding stars has been invented by[21]. BH optimization is particularly inspired by the nature or physics of BH, as well as its interaction with the surrounding stars. With the assumption that in a given iteration, a set of star is representative of the total number of solutions and each star is subjected to a pulling force towards the best solution representing BH. Then, a new set of solutions in the next iteration is generated by moving the stars toward the black hole, whereupon the star being within the predetermined distance to BH will render it swallowed and for alternative stars to be arbitrarily generated. This allows the algorithm to initiate an exploration in the searching space, rather than consuming the optimization time with an area fully discovered with solutions. In case of its implementation to solve a data clustering issue, it remains relevant despite performance evaluation showing that it is superior compared to other similar processes. Similarly, further enhancement for the approach will allow the discovery of powerful phenomenon in the solution space, while also making space for effectual clustering processing. In this perspective, the work of [21] can be developed from the objective function which does not assure the best possible accuracy, even when the cost is at the global optimum the original black hole algorithm suffers from weaknesses in exploration. Therefore, it requires too many reiterations to attain an optimum resolution. In recent years, the black hole algorithm and its modified versions have been used to solve engineering and optimization problems [26-37]. In this study, enhancing BH global search and resolving the issue of entrapment in the local minima have been undertaking by combining BH with levy flight. A Levy flight can be described as a type of arbitrary walk, namely generalized Brownian motion inclusive of non-Gaussian arbitrarily distributed step sizes for the distance moved. Different natural and man-made facts are explainable using Levy flight, which include fluid dynamics, earthquake analysis, fluorescent molecule diffusion, cooling behavior, noise, and more [38, 39]. Pereyra and Hadj have also opted for it in case of Ultrasound in Skin Tissue[40], while Al-teemy utilized it in Ladar Scanning[41]. Its role is also momentous in various computer science fields[42], with it being employed by Terdik and Gyres in designing Internet Traffic Models[41], Chen’s Delay and Disruption Tolerant Network, Sutantyo et al.’s Multi-Robot Searching procedure[42], and Rhee’s human mobility utilization [43]. Meanwhile, Yang and Deb [44, 45] opted for Levy flight distribution to generate a novel cuckoo in Cuckoo Search, alongside Yang’s introduction of an updated model of Firefly Algorithm-FA. The Levy-flight Firefly algorithm (LFA)[46] incorporates Firefly to unite Levy-flight with the search strategy so as to attain improved FA randomization. Lee and Yao’s Evolution Algorithm also developed four dissimilar states of parameters of Levy flight and 4 prospective solutions; the state offering the best results would be used for mutation procedure. Additionally, it was also utilized as a diversification tool in optimizing an ant colony.

In this paper, the long jumps have been undertaken via Levy distribution in order to ensure effectual use of the search space in comparison with BH. Previously investigated works have aimed to improve BH, whereby the current proposal calls for BH to perform random walks and global search. Thus, a Levy flight-based method combined with BH algorithm is proposed to resolve global optimization problems and data clustering problem. Levy flight, in particular, improves the global search capacity for the BH algorithm, preventing one to be stuck in local minima. Additionally, the proposed method enhances the global search ability of BH algorithm as per the new equation of star movements underlined. As BH algorithm is incapable of attaining the optimum results in a specific number of iterations, an efficient Levy-flight selection is imperative to avoid being stuck in local optimum as it results in improved global and local search capability concomitantly.

The remaining sections for this work will be arranged in the following manner: Section 2 will discuss some of the previously proposed research on data clustering. Then, the BH algorithm and proposed modified levy black hole algorithm is presented in Sections 3 and 4, respectively, whereas Section 5 outlines the experimental outcomes obtained. Finally, Section 6 will conclude the work succinctly.

II. OVERVIEW

A. The Problem of Data Clustering

Clustering can be described as an essential unsupervised classification approach characterized by the placement of a set of patterns or vectors (e.g. observations, data items, or feature vectors) into a multi-dimensional space in clusters or groups. This is achieved by utilizing similarity metrics between data

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2937021, IEEE Access

objects, whereby the similarity and dissimilarity of objects in the database are looked into using distance measurement [47]. The action is propelled by the idea of classifying a dataset provided using a specific number of clusters via distance minimization between objects of each cluster itself. Termed as cluster analysis, it is defined as the rearrangement of a body of patterns typically presented in two ways: 1) a vector of measurements, or 2) a point in a multi-dimensional space. This is done to obtain clusters that are characterized by the attribute of similarity [48, 49].

Clusters are oftentimes utilized for various applications, such as image processing, data statistical analysis, and medical imaging analysis, as well as other research fields of the science and engineering branch. Moreover, it is synonymous with statistical data analysis and known as a primary task for exploratory data mining in a multitude of fields, such as machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics. Figure 1 displays the difference between clusters that may be due to their shapes, sizes, and densities.

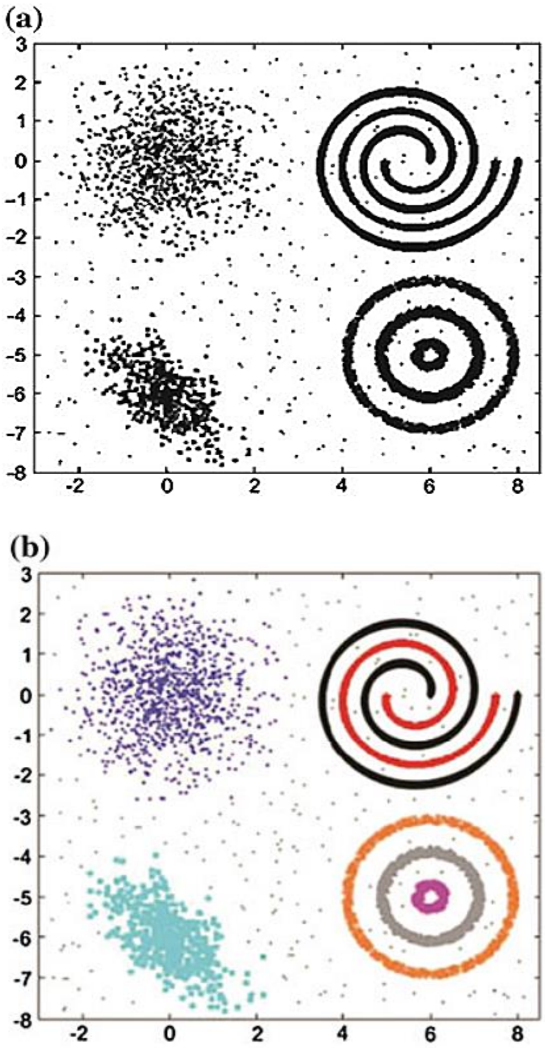

FIGURE 1. The difference between clusters a) Input data b) Fit Desired Clustering

However, noise present in the data may render cluster detection challenging, in which the ideal cluster is generally described as a compact and solitary set of points. Despite human beings having known to be proficient in cluster seeking in two and probably three dimensions, high-dimensional data calls for automatic algorithms. This fact, coupled with the unspecified number of clusters yet for data set provided, has continuously generated thousands of clustering algorithms in publication. In the context of pattern recognition, the data analysis section is particularly correlated with predictive modeling, in which training data is provided and the unknown test data's behavior is predicted. Such task is termed as learning.

An evaluation of the similarity of data objects requires the use of distance measurement. The problem may be framed as follows: given N records of data, each record is allocated to one of $K\ the$ clusters. Performing clustering has been carried out using different criteria that serve as an objective function for the process of optimization. One of the commonest attribute is minimizing the sum of squared Euclidean distance between each record and the center of the corresponding cluster as defined in [50] . This is displayed per equation (1) below.

$$F(O.Z) = \sum_{i=1}^{N} \sum_{j=1}^{K} W_{ij} \left\| O_i - Z_j \right\|^2, \quad (1)$$

Where N and K are the numbers of data records and the numbers of clusters, respectively. While $\left\| O_i - Z_j \right\|$ is the Euclidean distance between a data record O_i and the cluster center Z_j which is calculated as follows:

$$Z_j = \frac{1}{|N_j|} \sum_{i=1}^{N} W_{ij} O_i \quad (2)$$

Where N_j is the number of patterns in the ith cluster, $W_{ij}$the association weight of pattern O_i with cluster *j*. W_{ij} is 1 when O_i is allocated to cluster *j*, otherwise it is 0.

B. RELATED WORKS

The utilization of metaheuristic algorithms for the purpose of clustering problems has been discussed in various studies. This section is specifically driven to review metaheuristic-based clustering algorithms that are restricted to techniques that are linked to the proposed algorithm.

Van, D.M. and A.P. Engelbrecht. [51] had first proposed the data clustering approach using two means. The first is particle swarm optimization (PSO), whereby optimal centroids are found and utilized as a seed in the K-means algorithm. Meanwhile, the second approach entails the PSO usage in refining K-means formed clusters. Both have been tested and indicated their extensive potential.

Next, the Ant Colony Optimization (ACO) method has been discussed by Shelokar et al.[52]. It is characterized by the use of distributed agents mimicking the manner in which ants locate the shortest distance to a food source from their nest and return. The resulting observation indicates that it may be viable as an effectual heuristic for near-optimal cluster representation.

Senthilnath, Das, Omkar and Mani [53] comparatively studied three nature-inspired algorithms, namely GA, PSO, and Cuckoo Search (CS) on clustering problem. During the analysis CS was used with levy flight and the heavy-tail property of levy flight was exploited. The performance of these algorithms was evaluated on three standard datasets and one real-time multi-spectral satellite dataset while the results were analysed using various analytical techniques. The authors concluded that based on the given set of parameters, CS works better for most of the dataset due to the important role played by levy flight.

Singh and Sood [54] proposed a hybrid approach to show the swarm behaviour of clusters. They used a Krill herd algorithm to simulate the herding behaviour of each krill. The clusters were discovered using a density-based approach; it was also used to show the regions with sufficiently high-density krill clusters. The minimum distance from each krill to the food source and from high-density of herds were considered as the objective function of the krill movement. The movement of each krill is determined by the random diffusion and foraging movement.

An approach based on the combination of Levy flight with modified Bat algorithm to improve the clustering result has been proposed [55]. The proposed approach was tested on ten datasets and the experimental results showed that the proposed algorithm clusters the data objects efficiently. It also illustrates that it escapes from local optima and explores the search space effectively.

A new quantum chaotic cuckoo search algorithm (QCCS) was proposed by Boushaki, Kamel and Bendjeghaba [56] for data clustering. The superiority of CS over the conventional metaheuristics for clustering problems has been confirmed by various studies. However, all the cuckoos have a similar search pattern, and this may result to the premature convergence of the algorithm to local optima. Similarly, the convergence rate of the CS is sensitive to the randomly generated initial centroids seeds. Thus, the authors strived to extend the CS capabilities using nonhomogeneous update based on the quantum theory in a bid to tackle CS clustering problem in terms of the global search ability. They also replaced the randomness at the initialization step with a chaotic map to increase the efficiency of the search process and improve the convergence speed. An effective strategy was further developed for a proper management of the boundaries. The results of the experiments on six common real-life datasets show a significant superiority of the developed QCCS over eight recently developed algorithms, including, hybrid cuckoo search, genetic quantum cuckoo search, differential evolution, hybrid K-means, standard cuckoo search, improved cuckoo search, quantum particle swarm optimization, hybrid K-means chaotic PSO, differential evolution, and GA in terms of external and internal clustering quality.

A new version of Artificial Bee Colony (ABC) algorithm called History-driven Artificial Bee Colony (Hd-ABC) was proposed by Zabihi and Nasiri [57] by applying a memory mechanism to improve the performance of ABC. The proposed Hd-ABC uses a binary space partitioning (BSP) tree to memorize useful information of evaluated solutions. By the application of this memory mechanism, the fitness landscape can be approximated before the actual fitness evaluation. Fitness evaluation is a time and cost inefficient process in clustering problem, but the use of a memory mechanism has significantly reduced the number of fitness evaluations and facilitated the optimization process via the estimation of the solutions' fitness value instead of estimating the actual fitness values. The proposed data clustering algorithm was applied on 9 UCI datasets and 2 artificial datasets and both the statistical and experimental outcomes showed the proposed algorithm to perform better than the original ABC, its variants, and the other recent clustering algorithms.

III. METHODOLOGY

A. BLACK HOLE (BH) ALGORITHM

The design of the BH algorithm is rooted in the black hole occurrence and in the fundamental idea of a region of space hosting an extensive volume of mass concentrated within that no nearby object is capable of escaping from its gravitational pull. Upon falling into the phenomenon, one would be eliminated from the universe, light included.

The algorithm consists of two components: 1) the star movement, and 2) the star re-initialization crossing into the D-dimensional hypersphere around the black hole (i.e. termed as event horizon). It functions as follows: first, the $N + 1$ stars, $x_i \in R^D, i = 1, \dots, N + 1$ (where N is population size) are arbitrarily initialized in the search space. After their fitness evaluation, the best value is referred to as the black hole x_{BH}. Black hole is static; there is no movement until a better resolution is obtained by other stars. Thus, the number of individuals looking for the optimum value equals to N. Next, each generation has each star to move towards the black hole per the equation below:

$$x_i(t+1) = x_i(t) + rand \times \left(x_{BH} - x_i(t)\right) \qquad i = 1.2. \cdots N, \tag{3}$$

Where rand is a random number within an interval [0,1].

The BH algorithm also indicates that a star that founds itself too close the black hole beyond the event horizon will be eliminated. The radius of the event horizon (R) is described as follows:

$$R = \frac{f_{BH}}{\sum_{i=1}^{N} f_i}, \tag{4}$$

Where f_i and f_{BH} are the fitness values of black hole and ith star. N is the number of stars (candidate solutions).

In case of a distance that is less than R between a candidate solution and the black hole (best candidate), the particular candidate collapses and consequently, a new candidate is generated and arbitrarily disseminated in the search space. BH is commonly associated with a simple structure and ease of implementation, as well as a parameter-free algorithm. Its convergence to the global optimum occurs in all runs, whereas

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2937021, IEEE Access

other heuristic algorithms may encounter entrapment in the local optimum solutions [21, 58].

Despite excellent outcomes obtained when BH is utilized as a clustering technique, it is flawed by its weak balancing between exploration and exploitation capacities. A star may alter its direction if one of them finds a better solution compared to the solution for the current black hole, thereby transforming into a new black hole. Furthermore, the conceptualization of the event horizon has been made as the stars may display a relatively speedy convergence for the search space to be occupied by the black hole, due to the lack of exploration capabilities. However, it disallows the intensification of exploration or accumulation of knowledge regarding previously visited solution; it is simply a restart method subjected to each star individually [59]. Therefore, this study presents a modified BH algorithm in combination with levy flight for efficient data clustering.

B. Levy Flight BLACK HOLE (LBH) ALGORITHM

The proposed work aims to cluster and group the data objects in an efficient and effective manner. The method is founded upon the Levy flight in combination with the black hole (BH) algorithm for the purpose of global optimization and data clustering problems. Levy flight, in particular, enhances the global search capacity of the BH algorithm to prevent being stuck in local minima. Thus, the method improves the global search ability using a new equation for star movements. As the algorithm is incapable of finding optimum in a certain amount of iterations, Levy flight-based search is more efficient as it improves the local and global search concomitantly.

Some examples of Levy flight compared with the Brownian walk (random) have been displayed in Figure 2. After the first movements around a point, sudden jumps are encountered; it generates the simultaneous local and global search.

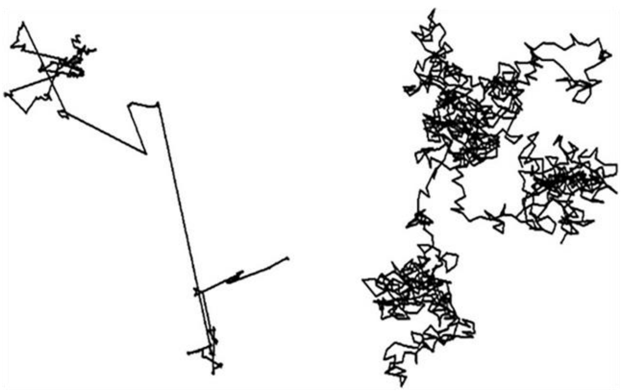

FIGURE 2 **The Levy flight and Brownian (random) walk**

Levy flight [60] can be defined as a type of arbitrary processes that is characterized by a jump size that adheres to the levy probability distribution function. Its name was derivative of a French mathematician named Paul Pierre Levy.

As a random walk, the steps in the Levy Flight are defined with respect to the step lengths. The step lengths have a given distribution probability and are drawn from a Levy distribution which is represented in Eq (5):

$$L(s) \sim |s|^{-1-\beta}, where\ \beta\ (0 < \ \beta \leq 2) \quad (5)$$

where β and s represents an index and the step length, respectively.

This study utilized a Mantegna algorithm for a symmetric Levy stable distribution to generate the sizes of the random steps. The term ‘symmetric’ in this concept implies that the step size will assume either a positive or negative value. The step length s in the Mantegna’s algorithm can be calculated thus:

$$s = \frac{u}{|v|^{1/\beta}} \quad (6)$$

where u and v are drawn from normal distributions; i.e.,

$$u \sim N(0, \sigma_u^2), \ \ v \sim N(0, \sigma_u^2) \quad (7)$$

Where

$$\sigma_u = \frac{\tau(1+\beta)\sin\frac{\pi\beta}{2}}{\tau[\left(\frac{1+\beta}{2}\right)\beta 2^{\frac{\beta-1}{2}}} \quad , \sigma_v = 1 \quad (8)$$

The distribution for s follows the anticipated Levy distribution for $|s| \geq |s_0|$, where s_0 represent the least step length and $\tau(.)$ represent the Gamma function which is estimated thus:

$$\tau(1+\beta) = \int_0^\infty t^\beta e^{-1} dt \quad (9)$$

The Levy distribution is used to generate the step sizes in the proposed technique. This is aimed at exploiting the search area. The step sizes are calculated thus:

$$step(t) = 0.01 \ \times \ s(t) \ \times \ rand(0,1) \quad (10)$$

where t represents an iteration counter, $s(t)$ is estimated as shown in Equation (6) using Levy distribution, while $rand(0,1)$ is a random value ranging from [0,1].

The step sizes in the Levy flights are too aggressive; this implies that they can often generate new solutions which are off the domain or on the boundary. Since the movement equation represented in the BH algorithm is a stochastic method search for new better positions within the search space, therefore, 0.01 multiplier is used in Equation (10) to reduce the step sizes when they get large. The positions of the stars are updated in the LBH as follows:

$$x_t(t+1) = x_t(t) + (step(t) \times \ (x_{BH} - x_t(t))) \quad (11)$$

where x_t is an individual star in iteration t while $step(t)$ is the actual step sizes generated using Equation (10). x_{BH} denotes the current best solution or the black hole.

Levy flight is characterized by an important parameter of β, whereby each star is a solution and an arbitrary number is produced as β between 0 and 2. Its different values may result in dissimilar outcomes. Therefore, larger values of β pose a higher likelihood to result in jumps to unexplored areas (i.e. higher exploration) and avoidance of being trapped in local optimums. However, smaller values will provoke the new positions to be viewed as near the obtained solutions (i.e. higher exploitation). The BH algorithm is particularly well-perceived for its excellent local search ability [59], but within the surround of the optimum point, it is characterized by a low convergence rate. This is due to higher exploitation rate compared to the exploration rate.

Hence, the suggested algorithm is designed in a manner that it allows the BH algorithm's local search ability, which will improve the method's efficiency in generating the optimal resolution and accelerating the convergence rate.

The proposed algorithm is named as Levy Flight Black Hole (LBH) algorithm and utilized to solve optimization and data clustering problems effectively. The pseudocode of LBH in Figure 3.

LBH Algorithm

1. **Input:** Dataset or Test Function, $MaxItr$, $PopSize$, $Upper$, $Lower$
2. **Output:** Best Solution X_{BH}
3. **Procedure:**
4. Define Objective Function $f(x_i)$
5. Initialize all the stars x_i in the population via uniform distribution
6. Evaluate the fitness value of each star X in the population via f
7. Set the best star in the population as Black Hole x_{BH}
8. ***While*** $itr \leq MaxItr$
9. ***For*** each star X_i ***in*** the population
10. Update the position of each star X_i via $eq. 11$
11. Check the boundaries of each star X_i
12. Evaluate the fitness value of the star X_i
13. Set the best star in the population as Black Hole x_{BH}
14. ***End For***
15. Calculate the event horizon via $eq. 4$
16. ***For*** each star X_i ***in*** the population
17. ***If*** X_i crosses the event horizon (R) ***Then***
18. Remove the star X_i
19. Generate a new star via $Step$ 5
20. ***End If***
21. ***End For***
22. Set the best star in the population as Black Hole X_{BH}
23. ***Loop***
24. Return X_{BH}

FIGURE 3. **The pseudocode of LBH algorithm**

IV. EXPERIMENTS AND RESULTS

The assessments were carried out on a personal computer (Core i7, 3.6 GHz, 16 GB of RAM, 64-bit Windows 10 Operating System) using MATLAB 2017a.

A. Evaluation of Benchmark Test Functions

As stated previously, the main contribution of this paper is to enhance the exploration of BH algorithm via Levy Flight. In order to further verify that the proposed algorithm has a better exploration than the standard BH, it has been evaluated on a set of unimodal and multimodal type of benchmark test functions in a multi-dimensional space as defined in [61-63]. The functions with their main characteristics in terms of Name, Dimensions (D), Upper and Lower Boundaries (*UB*, *LB*) and the value of the optimal solution (*Opt*) are stated in Table 1.

The comparison stage is done by benchmarking against nine well-known metaheuristics comprising of Big Bang–Big Crunch [64], Artificial Bees Colony (ABC)[65], Particle Swarm Optimization (PSO)[66], and Levy Firefly Algorithm [46](LFFA), Grey Wolf Optimizer (GWO)[19], Gravitational search algorithm (GSA) [67] , Bat algorithm (BA)[23], cat swarm algorithm (CSA)[68], and Black hole (BH)[21] respectively. The parameters settings for these algorithms are presented in Table 2.

The experiments for LBH and the other algorithms were executed in 30 different runs. The best, mean, error rate, and standard deviation were recorded and presented accordingly in Table 3. Additionally, the convergence curve of the searching has been generated for the first benchmark function and compared with other algorithms including the original BH algorithm. LBH has shown faster convergence curves for the first 100 iterations than the other algorithms. The convergence of BH by Levy flight (LBH) had enhanced the exploration ability of the algorithm and guided the stars towards better positions rate. Which means that the stars avoid the possibility of trapping in local optima. It can be seen that GWO and CSA algorithm have attained the second and the third place respectively, while the original BH attained the fourth place. Figure 4 shows the convergence and the 3D plot of sumsqaure (f_1).

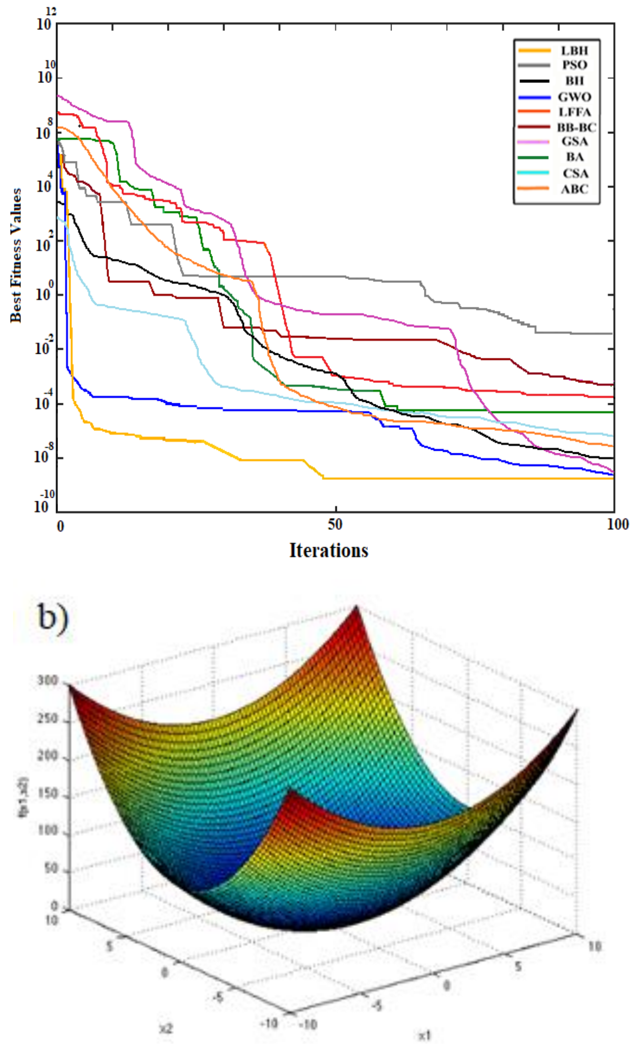

FIGURE 4. **The 3d plot of sumsqaure (f_1) a) The convergence analysis of LBH and other algorithms b) The 3D of f_1**

TABLE 1: Benchmark Test Functions

Fun	Name	Test	D	LB	UB	Opt
f_1	Sumsquare	$f_1(x) = \sum_{i=1}^{N} x_i^4$	30	-10	10	0
f_2	Rastrigin	$f_2(x) = \sum_{i=1}^{N}\{x_i^2 - 10\cos(2\pi x_i) + 10\}$	30	-5.12	5.12	0
f_3	Quartic	$f_3(x) = \sum_{i=1}^{n} ix_i^4 + random(0,1)$	30	-1.28	1.28	0
f_4	Ackley	$f_4(x) = -20e^{-0.02\sqrt{D^{-1}\sum_{i=1}^{D} x_1^2}} - e^{D^{-1}\sum_{i=1}^{D}\cos(2\pi x_i)} + 20 + e$	30	-32	32	0
f_5	Alpine No.1	$f_5(x) = \sum_{i=1}^{D} \lvert x_i \sin(x_i) + 0.1x_i \rvert$	30	-10	10	0
f_6	Griewank	$f_6(x) = \sum_{i=1}^{Dim} \frac{y_i^2}{4000} - \prod_{i=1}^{Dim} \cos\left(\frac{y_i}{\sqrt{i}}\right) + 1$	30	-600	600	0
f_7	Penalized	$f_7(x) = \sum_{i=1}^{Dim-1}(y_i - 1)^2 \times (1 + sin^2)(3\pi y_{i+1}) + (y_{Dim} - 1)^2(1 + sin^2(2\pi y_{Dim})) + sin^2(3\pi y_1)$	30	-50	50	0
f_8	Zakharov	$f_8(x) = \sum_{i=1}^{n} x_i^2 + (\frac{1}{2}\sum_{i=1}^{n} ix_i)^2 + (\frac{1}{2}\sum_{i=1}^{n} ix_i)^4$	30	-5	10	0
f_9	Sphere	$f_9(x) = \sum_{i=1}^{N} x_1^2$	30	-100	100	0

TABLE 2: Parameter Setting

Method	Parameters	Value
General	Swarm/Colony/Population Size	25
	Iterations	250
	Fitness function constant σ	0.999
	No. of Runs	30
LFFA	β_0	1.0
	γ	1.0
	α	0.2
	δ	0.96
PSO	ω	0.742
	c_1, c_2	1.42
GA	Migration Fraction	0.2
	Crossover Fraction	0.8
BA	Pulse Rate (r)	0.9
	Min Frequency (f_{min})	0
	Max Frequency (f_{max})	2
	Decrease Sound Loudness (a)	0.9
	Weighting Value (δ)	0.9
	Weighting Value(Φ)	0.1
ABC	No. of Source	Size / 2
	Limit	50
GWO	a	(2 –> 0.1)
CSA	c_1, c_2	2
	MD	0.1
	SMP	5
	SRD	0.4
	CDC	0.8
GSA	Gravitational Constant G_0	100
	β	20
	ε	2.22e-16

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2937021, IEEE Access

TABLE 3: The results of the standards algorithms and Levy black hole algorithm.

Fun	Statistics	BB-BC	ABC	PSO	LFFA	GWO	GSA	BA	CSA	BH	LBH
f_1	Best	4.1458	2.79E-16	2.13485	0.00774	**0.00000**	0.00156	2.267E+06	4.97E-04	3.34E-04	**0.00000**
	Mean	5.9475	2.72E-16	4.98451	0.21006	**0.00000**	0.02943	2.318E+06	0.00105	0.00348	**0.00000**
	Std. Div	2.1354	8.51E-12	3.94512	0.34752	**0.00000**	0.08790	5.125E+04	4.41E-04	3.12E-03	**0.00000**
f_2	Best	2.1049	1.40E-11	0.02448	4.94E-10	**0.00000**	0.90001	3.91E-09	**0.00000**	0.00845	**0.00000**
	Mean	3.3085	8.83E-13	2.15168	2.06E-07	**0.00000**	1.00043	4.24528	**0.00000**	0.08394	**0.00000**
	Std. Div	3.5478	2.76E-12	1.07664	5.18E-08	**0.00000**	0.90536	3.47563	**0.00000**	0.01945	**0.00000**
f_3	Best	3.45892	0.11531	1.3389	0.00409	0.00284	0.06348	0.10786	0.01741	0.02348	1.43E-04
	Mean	5.48953	0.19593	6.9606	0.02542	0.00379	0.08815	0.15314	0.02845	0.03154	9.15E-04
	Std. Div	0.83211	0.05549	0.6477	0.02312	0.00134	0.04413	0.00984	0.00148	0.00284	5.38E-04
f_4	Best	1.5829	0.02058	1.9877	0.0634	0.0692	2.86E-05	0.9900	1.2293	0.020580523	0.005829
	Mean	3.8331	0.15442	2.9439	1.9994	0.0366	0.0002763	0.9989	3.1853	0.069228159	0.038331
	Std. Div	1.0422	0.00000	0.037191	0.00013675	5.49E-10	0.556324	0.0497715	0.024199	0.019449	0.010422
f_5	Best	0.00064	0.00042	0.00425	0.00024	0.00116	0.00493	1.02E-04	5.82E-05	0.00481	4.91E-05
	Mean	1.06309	0.28568	2.67570	0.00029	0.10797	0.02171	0.33693	2.48E-03	0.08741	2.48E-04
	Std. Div	1.79308	0.62473	12.3490	0.00037	0.25769	0.00928	0.04030	0.00048	0.03847	0.00031
f_6	Best	**0.00000**	4.261E-06	0.15676	3.20E-07	**0.00000**	**0.00000**	3.33E-09	0.00019	0.001584	**0.00000**
	Mean	**0.00000**	0.0035	0.24208	1.51E-06	**0.00000**	**0.00000**	1.65E-05	0.00048	0.009612	**0.00000**
	Std. Div	**0.00000**	0.0067	0.09374	1.88E-06	**0.00000**	**0.00000**	1.99E-05	0.00082	0.084123	**0.00000**
f_7	Best	0.89765	0.47989	5.523E+08	**0.00000**	0.13732	15.3769	0.81675	0.14548	0.12245	**0.00000**
	Mean	0.56432	0.44998	7.899E+08	**0.00000**	0.23752	32366.20	1.34211	1.16473	0.26640	**0.00000**
	Std. Div	0.00318	0.00478	1.439E+08	**0.00000**	0.05676	59623.51	0.00671	0.40721	0.05789	**0.00000**
f_8	Best	4112.205	7726.247	3.55412	4021.309	1337.803	4214.467	3.55676	**0.00000**	13234.241	**0.00000**
	Mean	267.3249	8094.705	4.77746	277.7689	2035.742	345.7899	4.78767	**0.00000**	4.409E+16	**0.00000**
	Std. Div	189.7456	246.1136	0.85447	171.7327	3506.202	189.7867	0.89787	**0.00000**	1.5E+16	**0.00000**
f_9	Best	2.12461	0.00432	1.2945	0.00128	**0.00000**	0.04871	0.57843	0.00094	0.01745	**0.00000**
	Mean	3.98452	0.00645	2.7707	0.00300	**0.00000**	0.06643	0.76741	0.00845	0.04478	**0.00000**
	Std. Div	2.64871	0.03184	1.0831	0.00105	**0.00000**	0.00384	0.68817	0.05491	0.00648	**0.00000**

B. Evaluation based on Benchmark Datasets

The performance of the proposed algorithm for data clustering was evaluated using six datasets, namely: Iris, Wine, Glass, Cancer, Contraceptive Method Choice (CMC), and Vowel. Their respective characteristics are shown in Table 4. All data sets were sourced from the UCI machine learning laboratory.

- Iris dataset

The dataset consisted of 150 arbitrary samples of flowers having four features from the iris. They were differentiated into 3 groups of 50 instances, whereby each group represented a form of iris plant (Setosa, Versicolor and Virginica).

- Wine dataset

The dataset elucidated the quality of wine using the physicochemical properties, in which they were grown in the identical region in Italy but sourced from three cultivars, respectively. Each of the three types of wine was linked to 178 instances, with 13 numeric attributes representing the quantities of 13 components elicited in them.

- CMC dataset

The dataset was generated by TjenSien Lim, which is a subset of Indonesia's 1987 National Contraceptive Prevalence Survey. The sample size consisted of married women who were either not pregnant or not in the know of their pregnancy during the interview period. It featured the issue of predicting the recent contraceptive method choice (i.e. no use, long-term method, or short-term methods) according to a woman's demographic and socioeconomic attributes.

- Cancer dataset

The dataset was a representation of the Wisconsin breast cancer database, consisting of 683 instances having 9 components. They included: Clump Thickness, Cell Size Uniformity, Cell Shape Uniformity, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nuclei, and Mitoses. Each of the instances was possibly of one class, either benign or malignant.

- Glass dataset

The dataset consisted of 214 objects with nine features, which were: refractive index, sodium, magnesium, aluminum, silicon, potassium, calcium, barium, and iron. The data sampling was done using six groups of glass, which were: float processed building windows, non-float processed building windows, float-processed vehicle windows, containers, tableware, and headlamps.

- Vowel dataset

The dataset was comprised of 871 Indian Telugu vowel sounds, inclusive of three attributes that corresponded to the first, second and third vowel frequencies, as well as six overlapping classes.

The algorithm's performances were assessed and subjected to a comparison using two features:

- Sum of intra-cluster distances as an internal quality measure: The distance between each data object and the center of the corresponding cluster was calculated and totaled up, per equation (1). Generally, a smaller sum of intra-cluster distances was linked with a higher clustering quality. The sum of intra-cluster distances was also an assessment component for the fitness in this study.
- Error Rate (ER) as an external quality measure: The percentage of misplaced data objects as depicted in the equation below:

$$ER = \frac{Number\ of\ misplaced\ objects}{total\ number\ of\ objects\ within\ dataset} 100 \qquad (12)$$

TABLE 4. Main characteristics of the test datasets

Datasets	No. of classes	No. of features	No. of instances	The Size
Iris	3	4	150	50,50,50
Wine	3	13	178	59,71,48
CMC	3	9	1473	629,334,510
Cancer	2	9	683	444,178
Glass	6	9	214	70,17,76,13,9,29
Vowel	6	3	871	72,89,172,151,207,180

The performance showed by the proposed algorithm was compared against several heuristic methods previously explained in literature, such as K-means [48], PSO[69],ABC[70] , BAT[55], GSA[71], BB-BC[72], CS[56], GWO [73] and BH[21].

In contrast, LBH was compared against newer hybrid and modified meta-heuristics algorithms reported in the literature. They include: improved krill herd algorithm [74] hybrid clustering method using artificial bee colony and Mantegna levy distribution displayed in [75], a new quantum chaotic cuckoo search algorithm [56], Hd-ABC history-driven artificial bee colony [57] (ICAKHM) is regarded as a novel method which was designed based on a combination of K-harmonic means algorithm and a modified version of the imperialist competitive algorithm (ICA) presented in[76] and grey wolf optimizer with levy flight steps presented in [73].

Table 5 and Table 6 displayed the sum of intra-cluster distances and error rate using the standard meta-heuristics clustering algorithm and the hybrids and modified meta-heuristics algorithms alike to obtain a better comparison of the LBH.

In Table 5, a summary of intra-cluster distance and error rate is presented. The values for the best, average, worst, standard deviation and the error rate were calculated based on the simulation of each independent algorithm after 30 independent implementations. Best obtained values by algorithms are marked as bold for each dataset. The experimental results indicated that LBH better than BH and K-means. Furthermore, the suggested algorithm has the smallest standard deviation compared to other algorithms, which mean the LBH get to minimum value each time. Other algorithms is a little worse than LBH.

TABLE 5. The result obtained by LBH and standard algorithms on different data sets.

Dataset	Criteria	Kmeans	PSO	ABC	BA	GSA	BB-BC	CS	GWO	BH	LBH
	Best	97.32590	96.89428	N/A	97.433	96.68794	96.67648	97.98364	96.65826	96.65589	**96.5403**
	Average	106.5766	97.23280	96.65502	103.036	96.73105	96.76537	102.51332	99.12574	96.65681	**96.5622**
Iris	Worst	123.9695	97.89733	N/A	108.870	96.82463	97.42865	106.76087	N/A	96.66306	**96.5873**
	standard	12.938	0.347168	2.213958	3.410	0.02761	0.20456	2.182.24	N/A	0.001.73	**0.00014**
	Error rate	13.42	12.58	10.00	10.78	10.04	10.05	09.80	10.74	10.02	9.40
	Best	16,555.68	16,345.97	N/A	16,391.46	16,313.87	16,298.67	16,363.12	16,307.1	16,293.41	**16,291.99**
	Average	17,251.35	16,417.47	16,308.53	16,606.90	16,374.30	16,303.41	16,420.81	16,318.4	16,294.31	**16,292.99**
Wine	Worst	18,294.85	16,562.32	N/A	17,160.39	16,428.86	16,310.11	16,525.72	N/A	16,300.22	**16,296.89**
	standard	874.148	85,497.4	5.096923001	237.740	34,671.22	2,661.98	45,540.86	N/A	16,512.70	**0.90340**
	Error rate	31.14	28.52	28.76	28.92	29.15	28.52	29.10	29.56	28.47	28.40
	Best	5703.20000	5700.98500	N/A	5671.52600	5542.27631	5534.09483	5778.45388	N/A	5532.88323	**5531.99898**
	Average	5705.37000	5820.96500	5.584,630.1	5802.14400	5581.94502	5574.75174	5962.09604	N/A	5533.63122	**5532.29789**
CMC	Worst	5704.57000	5923.24900	N/A	5966.19000	5658.76293	5644.70264	6205.93042	N/A	5534.77738	**5532.58940**
	standard	1.033	46.95969	10.16857871	88.219	41.13648	39.43494	115.23954	N/A	0.59940	0.58878
	Error rate	54.48	54.49	57.68	56.00	55.67	54.52	57.18	N/A	54.39	54.35
	Best	2988.43000	2973.50000	N/A	3021.483000	2965.76394	2964.38753	3089.77652	2964.390	2964.38878	**2961.95000**
	Average	2988.99000	3050.04000	N/A	3107.125000	2972.66312	2964.38798	3200.79638	2964.395	2964.39539	**2963.90000**
Cancer	Worst	2999.19000	3318.88000	N/A	3250.525000	2993.24458	2964.38902	3476.06894	N/A	2964.45074	2988.43000
	standard	315.14560	110.8013	N/A	77.110	8.91860	**0.00048**	102.96476	N/A	0.00921	0.0072
	Error rate	04.39	05.25	N/A	03.79	03.74	03.70	04.94	03.65	03.70	3.65
	Best	215.73000	270.57000	N/A	232.00700	224.98410	223.89410	220.12580	265.8142	210.51549	**209.99689**
	Average	218.70000	275.71000	254.03500	241.91600	233.54329	231.23058	225.19820	302.0415	211.49860	**210.97180**
Glass	Worst	227.35000	283.52000	N/A	247.08500	248.36721	243.20883	227.02230	N/A	213.95689	**211.56990**
	standard	2.456	4.557134	10.107	5.059	6.13946	4.65013	5.6623	N/A	1.18230	**0.99869**
	Error rate	38.44	30.58	38.67	40.56	41.39	41.37	41.89	40.90	36.51	**30.50**
	Best	149,398.66	148,976.01	N/A	155,163.59	151,317.56	149,038.51	149,417.31	N/A	148,985.61	**148,965.64**
	Average	151,987.98	148,999.82	153,218.45	147,411.21	152,931.81	151,010.03	150,186.12	N/A	149,848.18	**149,466.52**
Vowel	Worst	162,455.69	149,121.18	N/A	160,783.94	155,346.69	153,090.44	150,841.40	N/A	153,058.98	**149,484.69**
	standard	3425.250	28.8134692	162,2703	3001.8245	2486.70285	1859.32353	1576.3697	N/A	1306.95375	**1297.64781**
	Error rate	43.57	41.92	42.87	42.55	42.39	41.89	42.41	N/A	41.65	**41.36**

In Iris dataset, LBH outperforms other algorithms of intra-cluster distance 96.5403 value and standard deviation 0.00014 in comparison to other algorithms. In the case of the Wine dataset, the proposed LBH algorithm obtained the optimum value of 16,291.99 which is remarkably superior compared to the other comparative algorithms. Similarly, upon comparison with the CMC dataset, the proposed LBH algorithm is also far better compared to the other algorithms, with the worst solution achieved at 5532.58940. However, it is still much better than the best solutions found by other algorithms. In case of the Cancer dataset, the proposed LBH algorithm's performance surpassed the K-means, PSO and GSA algorithms, but the BB–BC algorithm outcomes were superior compared to the proposed LBH in terms of standard deviation. For the Glass dataset, the suggested LBH algorithm obtained an average of 210.97180, whereas other algorithms failed to attain the solution at all. Meanwhile, the Vowel dataset was provided the best average solutions and standard deviation by the suggested LBH algorithm compared to the other algorithms. Therefore, the LBH offered better solution quality and smaller standard deviation in comparison with the other algorithms. LBH is capable of locating the optimal solutions as seen in a majority of the cases, while other algorithms may be trapped in local optima.

As per in Table 6, the proposed LBH obtained the best performance according to the average intra-cluster distances and error rate when subjected to a comparison with the remaining comparative algorithms. It also displayed better performance on all six datasets as opposed to the other comparative algorithms, in which a notable balance between exploitation and exploration enhanced the proposed LBH algorithms' performance.

On the Iris dataset, the standard deviation for the suggested LBH algorithm is 0.00014, which is significantly less than the other comparative algorithms. In contrast, the best solution is 96.5403 and the Worst is 96.5873, which is far superior compared to other algorithms. Furthermore, the Wine dataset indicated that the proposed LBH algorithm obtained the optimum value of 16,291.99, which surpassed the other algorithms.

The CMC dataset also yielded a proposed LBH algorithm that was far better compared to other algorithms, in which the worst solution attained is 5532.88940. This remained to be far superior to the best solutions obtained by the other algorithms. For the Cancer dataset, the proposed LBH best solutions are 2961.95000 and the average solution is 2963.90000, while the standard deviation is 0.00723. This was superior compared to ABCL, QCCS, HD-ABC, ICAKHM and EGWO.

Lastly, the Glass dataset obtained the best 199.86000 that was reached by the ICAKHM algorithm. Meanwhile, the Vowel dataset indicated that the suggested LBH algorithm provided the best average solutions 149,466.52. It passed sufficiently by yielding the best outcomes on almost all of the datasets and when compared to the other comparative algorithms. Thus, it proved that the suggested (LBH) was exceedingly effectual to resolve complex optimization problems, simply by the addition of new operators.

In addition to the previous presented comparison, the algorithms have been compared statistically based on Friedman test as well as the Iman–Davenport to determine whether there are significant differences in the results of the algorithms. Table 7 below shows the ranking of the algorithms based on them.

TABLE 7. The results of the statistical analysis tests

Test	Value	p- value	Results
Friedman test	11.79000	0.02538	Rejected
Iman-Davenport	5.15721	0.00214	Rejected

V. CONCLUSION

In this paper, Levy flight was combined with Black Hole algorithm to improve the clustering result. The suggested approach was subjected to testing on six datasets, whereby the experimental outcomes indicated that the proposed algorithm clustered the data objects efficiently. It also illustrated its escape from the local optima and exploration into the search space effectively. In the future, this work may be implemented to other applications, such as text document clustering for the purpose of clustering the set of documents effectively.

Table 6: The sum of intra-cluster distances and error rate obtained by LBH and modified algorithms on different data sets.

Datasets	Criteria	IKH	ABCL	QCCS	Hd-ABC	ICAKHM	EGWO	LBH
	Best	96.65550	N/A	96.65548	N/A	96.63620	96.65230	**96.5403**
	Average	96.65550	96.65550	96.65623	94.47	96.66640	99.12530	**96.5622**
Iris	Worst	96.65550	N/A	96.66771	N/A	96.69190	N/A	**96.5873**
	standard	9.8E − 06	1.351718	0.00266	N/A	0.01055	N/A	**0.00014**
	Error rate	9.78	10.45	09.43	0.0	11.23	9.76	9.40
	Best	16,292.21	N/A	16,292.26	N/A	16,293.90	16,292.15	**16,291.99**
	Average	16,294.30	16,295.30	16,293.26	16,280.96	16,295.60	16,292.43	**16,292.99**
Wine	Worst	16,292.84	N/A	16,294.34	N/A	16,296.94	N/A	**16,296.89**
	standard	0.706742	1.09745	0.71534	N/A	1.002372	N/A	**0903.40**
	Error rate	28.90	29.80	28.70	4.68	28.73	28.71	28.40
	Best	5693.720	N/A	5532.22476	N/A	5699.21830	N/A	**5531.99898**
	Average	5693.779	5533.7790	5532.71992	5692.75	5705.14850	N/A	**5532.29789**
CMC	Worst	5693.735	N/A	5535.29050	N/A	5721.17790	N/A	**5532.58940**
	standard	0.007975	0.85343	0.134	N/A	1.268275	N/A	0.58878
	Error rate	55.90	57.12	57.11	2.81	54.47	N/A	54.35
	Best	2964.387	N/A	2964.38951	N/A	2962.42000	2964.11000	**2961.95000**
	Average	2964.393	N/A	2964.41463	N/A	3022.81000	2964.49000	**2963.90000**
Cancer	Worst	2964.389	N/A	2964.49945	N/A	3150.15000	N/A	2988.43000
	standard	0.001258	N/A	0.02761	N/A	0.396	N/A	0.0072
	Error rate	3.69	N/A	03.51	N/A	4.27	3.75	3.65
	Best	210.2520	N/A	N/A	N/A	199.86000	214.42500	**209.99689**
	Average	222.8008	2.2009e+0	N/A	217.89	202.41000	242.43800	**210.97180**
Glass	Worst	215.9355	N/A	N/A	N/A	209.77000	N/A	**211.56990**
	standard	2.737919	4.6367333	N/A	N/A	0.26	N/A	**0.99869**
	Error rate	33.90	32.56	N/A	34.45	32.61	33.60	30.50
	Best	148,967.24	N/A	N/A	N/A	149,201.63	N/A	**148,965.64**
	Average	158,600.52	149,600.5	N/A	N/A	161,431.04	N/A	**149,466.52**
Vowel	Worst	150,172.42	N/A	N/A	N/A	165,804.67	N/A	**149,484.69**
	standard	1732.4516	1128.941	N/A	N/A	2746.041	N/A	**1297.64781**
	Error rate	41.56	41.90	N/A	N/A	41.98	N/A	41.36

REFERENCES

[1] M. Sarstedt, and E. Mooi, "Cluster analysis," *A concise guide to market research*, pp. 301-354: Springer, 2019.

[2] S. Arora, and I. Chana, "A survey of clustering techniques for big data analysis." pp. 59-65.

[3] C. C. Aggarwal, and C. K. Reddy, *Data clustering: algorithms and applications*: CRC press, 2013.

[4] X.-S. Yang, *Nature-inspired metaheuristic algorithms*: Luniver press, 2010.

[5] G. N. Kumar, B. V. Rao, D. D. Chowdary, and P. V. Sobhan, "Multi-Objective Optimal Power Flow Using Metaheuristic Optimization Algorithms With Unified Power Flow Controller to Enhance the Power System Performance," *Advancements in Applied Metaheuristic Computing*, pp. 1-33: IGI Global, 2018.

[6] M. N. Janardhanan, Z. Li, G. Bocewicz, Z. Banaszak, and P. Nielsen, "Metaheuristic algorithms for balancing robotic assembly lines with sequence-dependent robot setup times," *Applied Mathematical Modelling,* vol. 65, pp. 256-270, 2019.

[7] G. P. Gupta, and S. Jha, "Integrated clustering and routing protocol for wireless sensor networks using Cuckoo and Harmony Search based metaheuristic techniques," *Engineering Applications of Artificial Intelligence,* vol. 68, pp. 101-109, 2018.

[8] J. E. Diaz, J. Handl, and D.-L. Xu, "Integrating meta-heuristics, simulation and exact techniques for production planning of a failure-prone manufacturing system," *European Journal of Operational Research,* vol. 266, no. 3, pp. 976-989, 2018.

[9] Z. A. Al Sudani, S. Q. Salih, and Z. M. Yaseen, "Development of Multivariate Adaptive Regression Spline Integrated with Differential Evolution Model for Streamflow Simulation," *Journal of Hydrology*, 2019.

[10] A. A. Al-Musawi, A. A. Alwanas, S. Q. Salih, Z. H. Ali, M. T. Tran, and Z. M. Yaseen, "Shear strength of SFRCB without stirrups simulation: implementation of hybrid artificial intelligence model," *Engineering with Computers*, pp. 1-11, 2018.

[11] H. Tao, L. Diop, A. Bodian, K. Djaman, P. M. Ndiaye, and Z. M. Yaseen, "Reference evapotranspiration prediction using hybridized fuzzy model with firefly algorithm: Regional case study in Burkina Faso," *Agricultural water management,* vol. 208, pp. 140-151, 2018.

[12] K. N. Sayl, N. S. Muhammad, Z. M. Yaseen, and A. El-shafie, "Estimation the physical variables of rainwater harvesting system using integrated GIS-based remote sensing approach," *Water Resources Management,* vol. 30, no. 9, pp. 3299-3313, 2016.

[13] H. A. Ahmed, M. F. Zolkipli, and M. Ahmad, "A novel efficient substitution-box design based on firefly algorithm and discrete chaotic map," *Neural Computing and Applications*, pp. 1-10, 2018.

[14] A. A. Alzaidi, M. Ahmad, H. S. Ahmed, and E. A. Solami, "Sine-Cosine Optimization-Based Bijective Substitution-Boxes Construction Using Enhanced Dynamics of Chaotic Map," *Complexity,* vol. 2018, 2018.

[15] A. M. Taha, S.-D. Chen, and A. Mustapha, "NATURAL EXTENSIONS: BAT ALGORITHM WITH MEMORY," *Journal of Theoretical & Applied Information Technology,* vol. 79, no. 1, 2015.

[16] A. M. Taha, S.-D. Chen, and A. Mustapha, "Bat algorithm based hybrid filter-wrapper approach," *Advances in Operations Research,* vol. 2015, 2015.

[17] S. Q. Salih, A. A. Alsewari, B. Al-Khateeb, and M. F. Zolkipli, "Novel Multi-swarm Approach for Balancing Exploration and Exploitation in Particle Swarm Optimization." pp. 196-206.

[18] S. Q. Salih, and A. A. Alsewari, "Solving large-scale problems using multi-swarm particle swarm approach," *International Journal of Engineering & Technology,* vol. 7, no. 3, pp. 1725-1729, 2018.

[19] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software,* vol. 69, pp. 46-61, 2014.

[20] A. H. Gandomi, and A. H. Alavi, "Krill herd: a new bio-inspired optimization algorithm," *Communications in nonlinear science and numerical simulation,* vol. 17, no. 12, pp. 4831-4845, 2012.

[21] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Information Sciences,* vol. 222, pp. 175-184, 2013.
[22] X.-S. Yang, and S. Deb, "Cuckoo search via Lévy flights." pp. 210-214.
[23] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," *Nature inspired cooperative strategies for optimization (NICSO 2010)*, pp. 65-74: Springer, 2010.
[24] X.-S. Yang, "Engineering optimizations via nature-inspired virtual bee algorithms." pp. 317-323.
[25] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *simulation,* vol. 76, no. 2, pp. 60-68, 2001.
[26] E. Pashaei, and N. Aydin, "Binary black hole algorithm for feature selection and classification on biological data," *Applied Soft Computing,* vol. 56, pp. 94-106, 2017.
[27] H. Bouchekara, "Optimal power flow using black-hole-based optimization approach," *Applied Soft Computing,* vol. 24, pp. 879-888, 2014.
[28] R. Azizipanah-Abarghooee, T. Niknam, F. Bavafa, and M. Zare, "Short-term scheduling of thermal power systems using hybrid gradient based modified teaching–learning optimizer with black hole algorithm," *Electric Power Systems Research,* vol. 108, pp. 16-34, 2014.
[29] M. Nemati, and H. Momeni, "Black holes algorithm with fuzzy Hawking radiation," *International Journal of Scientific & Technology Research,* vol. 3, no. 6, pp. 85-88, 2014.
[30] H. R. Bouchekara, "Optimal design of electromagnetic devices using a black-hole-based optimization technique," *IEEE Transactions on Magnetics,* vol. 49, no. 12, pp. 5709-5714, 2013.
[31] M. Doraghinejad, and H. Nezamabadi-pour, "Black hole: a new operator for gravitational search algorithm," *International Journal of Computational Intelligence Systems,* vol. 7, no. 5, pp. 809-826, 2014.
[32] M. Eskandarzadehalamdary, B. Masoumi, and O. Sojodishijani, "A new hybrid algorithm based on black hole optimization and bisecting k-means for cluster analysis." pp. 1075-1079.
[33] S. Yaghoobi, S. Hemayat, and H. Mojallali, "Image gray-level enhancement using Black Hole algorithm." pp. 1-5.
[34] E. Pashaei, M. Ozen, and N. Aydin, "An application of black hole algorithm and decision tree for medical problem." pp. 1-6.
[35] K. Premalatha, and R. Balamurugan, "A nature inspired swarm based stellar-mass black hole for engineering optimization." pp. 1-8.
[36] K. Lenin, B. R. Reddy, and M. S. Kalavathi, "Dwindling of active power loss by enhanced black hole algorithm," *Int. J. Res. Electron. Comm. Tech.,* vol. 1, no. 4, pp. 11-15, 2014.
[37] D. Rodrigues, C. C. O. Ramos, A. N. De Souza, and J. P. Papa, "Black hole algorithm for non-technical losses characterization." pp. 1-4.
[38] Y. Chen, "Research and simulation on Levy flight model for DTN." pp. 4421-4423.
[39] G. M. Viswanathan, V. Afanasyev, S. Buldyrev, E. Murphy, P. Prince, and H. E. Stanley, "Lévy flight search patterns of wandering albatrosses," *Nature,* vol. 381, no. 6581, pp. 413, 1996.
[40] M. A. Pereyra, and H. Batatia, "A Levy flight model for ultrasound in skin tissues." pp. 2327-2331.
[41] G. Terdik, and T. Gyires, "Lévy flights and fractal modeling of internet traffic," *IEEE/ACM Transactions on Networking,* vol. 17, no. 1, pp. 120-129, 2009.
[42] D. K. Sutantyo, S. Kernbach, P. Levi, and V. A. Nepomnyashchikh, "Multi-robot searching algorithm using Lévy flight and artificial potential field." pp. 1-6.
[43] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong, "On the levy-walk nature of human mobility," *IEEE/ACM transactions on networking (TON),* vol. 19, no. 3, pp. 630-643, 2011.
[44] M. F. Tasgetiren, Y.-C. Liang, M. Sevkli, and G. Gencyilmaz, "A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem," *European journal of operational research,* vol. 177, no. 3, pp. 1930-1947, 2007.

[45] X.-S. Yang, and S. Deb, "Multiobjective cuckoo search for design optimization," *Computers & Operations Research,* vol. 40, no. 6, pp. 1616-1624, 2013.

[46] X.-S. Yang, "Firefly algorithm, Levy flights and global optimization," *Research and development in intelligent systems XXVI*, pp. 209-218: Springer, 2010.

[47] M. R. Anderberg, *Cluster analysis for applications*, Office of the Assistant for Study Support Kirtland AFB N MEX, 1973.

[48] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern recognition letters,* vol. 31, no. 8, pp. 651-666, 2010.

[49] J. A. Hartigan, "Clustering algorithms," 1975.

[50] M. B. Dowlatshahi, and H. Nezamabadi-Pour, "GGSA: a grouping gravitational search algorithm for data clustering," *Engineering Applications of Artificial Intelligence,* vol. 36, pp. 114-121, 2014.

[51] X. Xiao, E. R. Dow, R. Eberhart, Z. B. Miled, and R. J. Oppelt, "Gene clustering using self-organizing maps and particle swarm optimization." p. 10 pp.

[52] P. Shelokar, V. K. Jayaraman, and B. D. Kulkarni, "An ant colony approach for clustering," *Analytica Chimica Acta,* vol. 509, no. 2, pp. 187-195, 2004.

[53] J. Senthilnath, V. Das, S. Omkar, and V. Mani, "Clustering using levy flight cuckoo search." pp. 65-75.

[54] V. Singh, and M. M. Sood, "Krill Herd clustering algorithm using dbscan technique," *Int. J. Comput. Sci. Eng. Technol,* vol. 4, no. 03, pp. 197-200, 2013.

[55] R. Jensi, and G. W. Jiji, "MBA-LF: A NEW DATA CLUSTERING METHOD USING MODIFIED BAT ALGORITHM AND LEVY FLIGHT," *ICTACT Journal on Soft Computing,* vol. 6, no. 1, 2015.

[56] S. I. Boushaki, N. Kamel, and O. Bendjeghaba, "A new quantum chaotic cuckoo search algorithm for data clustering," *Expert Systems with Applications,* vol. 96, pp. 358-372, 2018.

[57] F. Zabihi, and B. Nasiri, "A Novel History-driven Artificial Bee Colony Algorithm for Data Clustering," *Applied Soft Computing,* vol. 71, pp. 226-241, 2018.

[58] S. Kumar, D. Datta, and S. K. Singh, "Black hole algorithm and its applications," *Computational intelligence applications in modeling and control*, pp. 147-170: Springer, 2015.

[59] A. P. Piotrowski, J. J. Napiorkowski, and P. M. Rowinski, "How novel is the "novel" black hole optimization approach?," *Information Sciences,* vol. 267, pp. 191-200, 2014.

[60] A. V. Chechkin, R. Metzler, J. Klafter, and V. Y. Gonchar, "Introduction to the theory of Lévy flights," *Anomalous transport: Foundations and applications,* vol. 49, no. 2, pp. 431-451, 2008.

[61] N. S. Jaddi, J. Alvankarian, and S. Abdullah, "Kidney-inspired algorithm for optimization problems," *Communications in Nonlinear Science and Numerical Simulation,* vol. 42, pp. 358-369, 2017.

[62] L. Zhang, L. Shan, and J. Wang, "Optimal feature selection using distance-based discrete firefly algorithm with mutual information criterion," *Neural Computing and Applications,* vol. 28, no. 9, pp. 2795-2808, 2017.

[63] B. Niu, Y. Zhu, X. He, and H. Wu, "MCPSO: A multi-swarm cooperative particle swarm optimizer," *Applied mathematics and computation,* vol. 185, no. 2, pp. 1050-1062, 2007.

[64] O. K. Erol, and I. Eksin, "A new optimization method: big bang–big crunch," *Advances in Engineering Software,* vol. 37, no. 2, pp. 106-111, 2006.

[65] D. Karaboga, and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of global optimization,* vol. 39, no. 3, pp. 459-471, 2007.

[66] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas, "Standard particle swarm optimisation 2011 at cec-2013: A baseline for future pso improvements." pp. 2337-2344.

[67] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences,* vol. 179, no. 13, pp. 2232-2248, 2009.

[68] S.-C. Chu, P.-W. Tsai, and J.-S. Pan, "Cat swarm optimization." pp. 854-858.

[69] J. Kennedy, and R. Eberhart, "Particle swarm optimization inProceedings of IEEE

International Conference on Neural Networks," *Piscataway December*, 1995.
[70] T. A. Runkler, "Ant colony optimization of clustering models," *International Journal of Intelligent Systems,* vol. 20, no. 12, pp. 1233-1251, 2005.
[71] A. Hatamlou, S. Abdullah, and H. Nezamabadi-Pour, "Application of gravitational search algorithm on data clustering." pp. 337-346.
[72] A. Hatamlou, S. Abdullah, and M. Hatamlou, "Data clustering using big bang–big crunch algorithm," *Innovative Computing Technology*, pp. 383-388: Springer, 2011.
[73] V. Kumar, J. K. Chhabra, and D. Kumar, "Grey wolf algorithm-based clustering technique," *Journal of Intelligent Systems,* vol. 26, no. 1, pp. 153-168, 2017.
[74] R. Jensi, and G. W. Jiji, "An improved krill herd algorithm with global exploration capability for solving numerical function optimization problems and its application to data clustering," *Applied Soft Computing,* vol. 46, pp. 230-245, 2016.
[75] H. Ghafarzadeh, and A. Bouyer, "An Efficient Hybrid Clustering Method Using an Artificial Bee Colony Algorithm and Mantegna Lévy Distribution," *International Journal on Artificial Intelligence Tools,* vol. 25, no. 02, pp. 1550034, 2016.
[76] A. Bouyer, and A. Hatamlou, "An efficient hybrid clustering method based on improved cuckoo optimization and modified particle swarm optimization algorithms," *Applied Soft Computing,* vol. 67, pp. 172-182, 2018.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2937021, IEEE Access