Knowledge-Based Systems 248 (2022) 108833

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Efficient text document clustering approach using multi-search Arithmetic Optimization Algorithm

Laith Abualigah [a,*], Khaled H. Almotairi [b], Mohammed A.A. Al-qaness [c], Ahmed A. Ewees [d], Dalia Yousri [e], Mohamed Abd Elaziz [f,g,h], Mohammad H. Nadimi-Shahraki [i,j]

[a] Faculty of Computer Sciences and Informatics, Amman Arab University, 11953 Amman, Jordan
[b] Computer Engineering Department, Umm Al-Qura University, Makkah, Saudi Arabia
[c] State Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China
[d] Department of Computer, Damietta University, Damietta 34517, Egypt
[e] Department of Electrical Engineering, Faculty of Engineering, Fayoum University, Fayoum, Egypt
[f] Faculty of Computer Science and Engineering, Galala University, Suze 435611, Egypt
[g] Artificial Intelligence Research Center (AIRC), Ajman University, Ajman 346, United Arab Emirates
[h] Department of Mathematics, Faculty of Science, Zagazig University, Zagazig 44519, Egypt
[i] Faculty of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran
[j] Big Data Research Center, Najafabad Branch, Islamic Azad University, Najafabad, Iran

ARTICLE INFO

Article history:
Received 1 February 2022
Received in revised form 11 April 2022
Accepted 14 April 2022
Available online 25 April 2022

Keywords:
Arithmetic Optimization Algorithm (AOA)
Opposition-based learning
Levy flight
Text clustering
CEC2019 problems

ABSTRACT

Text document clustering is to divide textual contents into clusters or groups. It received wide attention due to the vast amount of daily data from the Web. In the last decade, Meta-Heuristic (MH) techniques have been adopted to solve clustering problems. Motivated by that, the authors introduce a reliable version of the newly developed MH algorithm called Arithmetic Optimization Algorithm (AOA). Math arithmetic operators inspire the AOA: multiplication, subtraction adding, and division. The AOA showed good performance in several global problems; nonetheless, it suffers from entrapment in local optima in complicated and high dimensional problems. Therefore, this paper proposes an improved version of AOA for the text document clustering problem. The Improved AOA (IAOA) introduces an integration between Opposition-based learning (OBL) and Levy flight distribution (LFD) with AOA to tackle the limitations of the traditional AOA. The IAOA is examined with different UCI datasets for the text clustering problems and assessed with a set of CEC2019 benchmark functions as a global optimization algorithm with extensive comparison to existing optimization algorithms. Overall, experimental results show the superiority of the proposed IAOA compared to several optimization algorithms. Moreover, the proposed IAOA is compared with twenty-one state-of-the-art methods using thirty-one benchmark text datasets, and the results proved the superiority of the proposed IAOA.

1. Introduction

With the rapid growth of internet technology, enormous amounts of text on web pages and intelligent applications need to be processed using complex analysis tools for text mining [1]. Text document clustering is an integral approach applied in the text mining domain, as in various machine learning and pattern recognition applications. It can be defined as a process that categorizes a set of text documents into hierarchies of semantic clusters [2]. Generally, text document clustering applications include information retrieval, document organization, and automatic topic extraction [3]. Clustering also facilitates various applications, such as text classification, text categorization, search engines, and image recognition [4,5].

Text documents clustering is an optimization problem [6], therefore, in recent years various Meta-Heuristics (MH) optimization algorithms have been proposed to solve this NP hard problem [7]. For example, Genetic algorithm [8], african vultures optimization algorithm [9], firefly algorithm [10], artificial bee colony (ABC) [11], firefly algorithm [12], dwarf mongoose optimization algorithm [13], krill herd algorithm [14], gravitational search algorithm [15], differential evolution algorithm [16], farmland fertility [17], harris hawk algorithm [18], gray wolf optimizer [19], moth-flame optimization algorithm [20], starling murmuration optimizer [21], cuckoo search algorithm [22],

* Corresponding author.
E-mail addresses: Aligah.2020@gmail.com (L. Abualigah), khmotairi@uqu.edu.sa (K.H. Almotairi), alqaness@whu.edu.cn (M.A.A. Al-qaness), ewees@du.edu.eg (A.A. Ewees), day01@fayoum.edu.eg (D. Yousri), abd_el_aziz_m@yahoo.com (M.A. Elaziz), nadimi@iaun.ac.ir (M.H. Nadimi-Shahraki).

https://doi.org/10.1016/j.knosys.2022.108833
0950-7051/

multi-verse optimizer (MVO) [23], ebola Optimization Search Algorithm [24], sine cosine algorithm [25], tree seed algorithm [26], particle swarm optimization [27], aquila optimizer [28], spotted hyena optimizer [29], moth flame optimization algorithm [30], symbiotic organisms algorithm [31], reptile search algorithm [32], and others [33–35].

However, individual optimization MH algorithms may face critical challenges in searching for an optimal solution, such as slow convergence and entrapment in local optima. To solve these challenges, in recent years, the hybridization concept was utilized to leverage the advantages of two hybrid search techniques (i.e., MH algorithms) to avoid the shortcomings of the individual one. Thus, several hybrid methods have been proposed in the literature [36,37] as follows. Purushothaman et al. [38] proposed a hybrid of GWO and Grasshopper Optimization Algorithm (GOA). The proposed approach, called GWO-GOA, was applied for FS and text clustering and was evaluated with eight datasets. Overall, it showed better performance than the traditional versions of GOA and GWO. It also minimized the computational time cost and improved reliability.

Aljarah et al. [39] proposed a text clustering approach using a hybrid GWO and tabu search (TS) algorithm. The TS is employed to enhance the searchability of the GWO. The proposed GWOTS were evaluated using different clustering datasets. It showed superior performance than the traditional GWO; however, its computational time is high. Kushwaha and Pant [2] proposed a new version of the PSO, called link-based PSO (LBPSO). They applied a new neighborhood strategy for the LBPSO to select features. The proposed approach is applied for text clustering, which exceeds the traditional PSO and challenges using global best updating mechanisms for feature selection. Thus, the particles of the LBPSO are updated using the neighbor's best position instead of the best global one. The LBPSO was evaluated with different datasets, and it showed significant performance compared to traditional PSO. Abasi et al. [23] proposed a new modified MVO, called link-based MVO, called LBMVO. The LBMVO overcomes the limitations of the traditional MVO by enhancing its exploitation phase. The evaluation experiments confirmed the superiority of the LBMVO over the traditional MVO; also, it showed better performance than several existing clustering approaches. In [3], a new text document clustering method was proposed by combining the Spectral Clustering algorithm and PSO. The proposed approach, SCPSO, achieved better performance compared to several clustering methods, including the original PSO, expectation–maximization method, and spherical K-means.

A hybrid intelligence method using a K-means algorithm for text clustering is presented in [40]. First, ten benchmark functions are used to evaluate the hybrid fruit-fly optimization technique. The suggested technique is then tested on six different benchmark text datasets. According to the conducted experiments, the suggested methodology is resilient and superior to existing methods. The interplay between topic modeling and document clustering is demonstrated using a new unsupervised technique [41]. The proposed method uses Bayesian design to suit and posterior inference to seamlessly combine and carry out the two objectives. Compared to state-of-the-art rivals and customized baselines, the experimental findings show that it is more successful at splitting text collections and determining their meanings. Rahnema et al. [42] proposed an enhanced version of the ABC algorithm using the WOA algorithm for data clustering. The operators' WOA is applied to enhance ABC's exploration, and random memory is employed to increase the convergence. The proposed approach ABCWOA was evaluated with different UCI datasets and compared to several existing clustering methods. It showed significant performance in all tests. Also, in [43], a multi-objective ABC algorithm was proposed depending on decomposition for multi-document text summarization. Mosa [44] proposed a hybrid optimization approach of PSO and GSA for a multi-optimization task in social media text mining. The PSO is applied to enhance the local search of the traditional GSA and slow its fast convergence.

This paper proposes a novel clustering method for text documents using a modified version of a recently proposed MH algorithm. The Arithmetic Optimization Algorithm (AOA) is a new MH proposed by [45]. Like other MH algorithms, AOA has two cores, exploration and exploitation, inspired by math arithmetic operators, such as multiplication, division, subtraction, and adding. It is a population-based and gradient-free algorithm. As described in [45], AOA showed superior performance in solving various optimization algorithms. However, the traditional AOA suffers from entrapment in local optima in case of complicated and high-dimensional problems. Therefore, the Opposition-based learning (OBL) [46] and Levy flight distribution (LFD) is utilized to improve the traditional AOA and to build a new version called IAOA. The main idea of the IAOA is to apply the two search techniques, OBL and LFD, to enhance search performance. The OBL is first employed to increase the rate of convergence. Then, the LFD is employed to optimize the discovery search. We evaluate the proposed methods with ten datasets for text clustering, and a set of CEC2019 benchmark functions as a global optimization algorithm is used. The new proposed IAOA confirmed its high performance with extensive comparisons to existing MH algorithms.

To sum up, this paper presents the following contributions:

1. A novel Meta-Heuristic optimization approach based on Arithmetic Optimization Algorithm (AOA) is proposed for text document clustering.
2. The original AOA's main cores are enhanced via merging the operator of Levy flight distribution to improve the exploration ability of AOA and the operator of the Opposition-based learning to promote the convergence speed of the algorithm.
3. The proposed modifications are assessed with a set of ten challenging CEC2019 benchmark functions.
4. The Improved AOA (IAOA) is applied to a set of text document clustering datasets and compared with other MH techniques using comprehensive statistical metrics.

Finally, this paper is presented as follows. Problem definitions and procedures are described in Section 2. Section 3 presents the proposed IAOA, and Section 4 presents the evaluation experiments and results. The conclusion and future works are highlighted in Section 5.

2. Background/ methods

This section presents the problem definitions and procedures. In addition, conventional AOA, opposition-based learning (OBL), and Levy Flight distribution (LFD).

2.1. Problem definitions and procedures

This section gives the text document clustering problems and its mathematical formulations as follows.

2.1.1. Problem descriptions and formulations

The text document set (D) is divided or partitioned into K number of clusters, which is given in advance, where D is performed as a vector of n documents $D = (d1, d2, \ldots\ldots, di, \ldots\ldots, dn)$, $d1$ denotes to the document number 1 in D, i is the number of the i_{th} document, and n is the number of all given documents in D [11,47]. One centroid is determined for each cluster c, so, the i_{th} centroid of the i_{th} cluster is represented by a vector of terms weights $ci = (ci1, ci2, \ldots, cij, \ldots, cit)$. Where $ci1$ is the first position of the i_{th} cluster centroid; this position contains the average weighting score of the first term in the i_{th} cluster centroid [14]. Moreover, t is the total number of all unique terms (features) given in the centroids vectors. Notably, the similarity measure is used to assign each document to the similar cluster centroid [48,49].

2.1.2. Text document preprocessing

For text representation, text clustering methods require preprocessing. Natural Language Processing (NLP) is a human–computer language communications system [1]. This system is essential and is a critical step in text mining. It is used to qualify machines to extract meanings from human language. These processes specify the number of tokens as terms called the simple text preprocessing used to extract document attributes [50]. The preprocessing steps are classified into three steps; tokenization, stop word elimination, stemming, and [51].

1. Tokenization The process of breaking statements into words, called tokens, is tokenization, perhaps missing certain characters simultaneously, such as punctuation [4]. These tokens are generally referred to as terms or phrases, but a distinction between tokens is necessary. A token is an example of a sequence of characters classified as a functional semantic unit in a text. A type is the set of all tokens, including the same sequence of characters. A word is a category that is included in the vocabulary of the information retrieval system [52].
2. Stop words removal Stop words are typical common words, such as in "an", "we", "some", "beside", "between", "but", "that", and other similar words in the text document that are mostly used and small functional words. As they usually have high frequency, these words need to be deleted from text records, which reduces the text clustering technique's efficiency. The list[1] of stop words includes around 571 words [53].
3. Stemming Stemming is how inflected words are simplified to their word stem (root). The stem approach is not the same as the root morphological process; it is typically used to map words to the same stem, even if it is not a valid root. The most popular stemming tool used in text mining, [54], is Porter.[2] For Natural Language Text Processing, all text studies recommended come from Python NLTK Demos.[3] For example, the "wait" stem holds similar words like wait, waits, waited, and waiting.
4. Document representation
   The Vector Space Model (VSM) is a successful representation that uses the standard text format to describe text documents. In the early 1970s, it was launched [55]. Each text is represented as a vector of terms weighting scores to promote similarity estimation. In order to increase the efficiency of the text analysis method and to minimize the total time cost [56], each word in the set represents a dimension of the weighted value. In several text mining contexts, VSM is used, such as information retrieval [57], text classification [58], text feature selection [59], and text clustering [60].
   The term weighting is represented by the Vector Space Model (VSM) in which text documents are shown in the regular format [61], as seen in Eq. (1). Eq. (1) is a general standard format representing n documents and t terms using VSM. Each document is represented by this method as a vector, as seen in Eq. (2) [62].

$$VSM = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,(t-1)} & w_{1,t} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{(n-1),1} & w_{(n-1),2} & \cdots & \cdots & w_{(n-1),t} \\ w_{n,1} & w_{n,2} & \cdots & w_{n,(t-1)} & w_{n,t} \end{bmatrix} \quad (1)$$

$$d_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,j}, \ldots, w_{i,t}), \quad (2)$$

2.1.3. Term weighting

Term weighting is an efficient computational mathematical step in order to consider the weighting of document words (terms or features) according to the term frequency for text document clustering operations [63,64]. The term frequency-inverse document frequency (TF-IDF) is a standard weighting scheme utilized in the text-mining domain for document representation [53]. The term weight is calculated by Eq. (3), as follows:

$$w_{i,j} = tf(i,j) * log(n/df(j)), \quad (3)$$

where, $w_{i,j}$ denotes to the weight value of the j_{th} term in the i_{th} document, *tf(i,j)* denotes to the occurrences or term frequency of the j_{th} term in the i_{th} document, n is the total number of all given documents in dataset (D), and *df(j)* is the number of documents that bearing the term number j.

2.1.4. Solution representation

Clustering of text documents is an optimization problem implemented based on optimization algorithms (i.e., AOA). To overcome this problem, optimization algorithms deal with multiple solutions. The suggested solution for resolving the text clustering is illustrated by each solution (row) [65]. The solution is defined as a n length vector representing each document's location in the D; each position corresponds to a document belonging to a cluster. Fig. 1 shows a description of the solution, where the i_{th} position of the solution corresponds to the decision regarding the i_{th} document. If the number of the given clusters is K, so the decision values in the range $(1, \ldots, K)$ correlates to each location of the solution. Notably, the number of clusters is given in advance.

In Fig. 1, ten documents that belong to four clusters are given. The solution length refers to the number of the given documents. Each decision refers to a number in the range of the number of clusters (1–4). In this example, documents 1 and 9 belong to the first cluster as label 1. Meanwhile, documents 2, 7, and 8 belong to the same cluster as label 2. The fourth cluster contains documents number 3 and 4.

2.1.5. Fitness function

Fitness function value is determined to evaluate each solution based on its positions values [14]. Each set of documents has a K centroids $C = (c_1, c_2, \ldots, c_j, \ldots, c_K)$, where c_j refers to the centroid of the j_{th} cluster. The fitness value for each solution is calculated by the average similarity of documents to the cluster centroids (ASDC) [66,67], as presented in Eq. (4).

$$ASDC = \left[\frac{\sum_{j=1}^{K} \left(\frac{\sum_{i=1}^{n} Cos(d_i, c_j)}{m_i} \right)}{K} \right], \quad (4)$$

where K refers to the number of clusters and m_i is the number of documents that belong to the i_{th} cluster. $Cos(d_i, c_i)$ is the similarity between the centroid of cluster j and the document number i.

[1] http://www.unine.ch/Info/clef/
[2] Porter stemmer. website at http://tartarus.org/martin/PorterStemmer/.
[3] http://text-processing.com/demo/

Fig. 1. Solution representation.

Each solution representation must denote binary matrix $a_{i,j}$ of size $n * K$ to compute the clusters centroid, as shown in Eq. (5).

$$a_{i,j} = \begin{cases} 1, & \text{if } d_i \text{ is assigned to the } j_{th} \text{ cluster} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

Eq. (6) is used to compute the j_{th} cluster centroid, which is represented as a vector $c_j = (c_1, c_2, \ldots, \ldots, c_t)$ [11].

$$c_j = \frac{\sum_{i=1}^{n} a_{ij}(w_{ij})}{\sum_{i=1}^{n} a_{ij}}, \quad (6)$$

where a_{ij} represents a binary matrix that contains the clustered data (see Eq. (5)), w_{ij} is the j_{th} term weighting of the document i, and n is the number of documents.

2.2. Arithmetic Optimization Algorithm (AOA)

AOA is a new optimizer proposed by Abualigah et al. in 2021 [45]. The exploration (diversification) and exploitation (intensification) stages of the conventional AOA are defined in this segment, which are inspired by the Arithmetic operators in math (i.e., Multiplication (M '$\times$'), Division (D '$\div$'), Subtraction (S '$-$'), and Addition (A '$+$'). AOA is a gradient-free and population-based approach, so it is being used to solve small or large optimization problems as per the required formulation.

2.2.1. Inspiration

The core inspiration for the AOA stems from arithmetic operators' use to solve the problems with arithmetic. The following subsections will describe arithmetic operators (i.e., multiplication, division, subtraction, and addition) and their effect on the AOA optimizer.

2.2.2. Initialization phase

In AOA, the evaluation process begins with a randomized set of candidate solutions (X) as can be seen in Matrix (7), and the best president solution in each evaluation (iteration) is assumed to be the best option or almost the equilibrium so far.

$$X = \begin{bmatrix} x_{1,1} & \cdots & \cdots & x_{1,j} & x_{1,n-1} & x_{1,n} \\ x_{2,1} & \cdots & \cdots & x_{2,j} & \cdots & x_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-1,1} & \cdots & \cdots & x_{N-1,j} & \cdots & x_{N-1,n} \\ x_{N,1} & \cdots & \cdots & x_{N,j} & x_{N,n-1} & x_{N,n} \end{bmatrix} \quad (7)$$

The search stage should be selected before the AOA begins running (i.e., exploration or exploitation). Therefore the Math Optimizer Accelerated (MOA) function is a measured by Eq. (8), which is used in the following search phases.

$$MOA(C\_Iter) = Min + C\_Iter \times \left(\frac{Max - Min}{M\_Iter}\right) \quad (8)$$

Where MOA(C Iter) represents the tth repeat function value that is determined by Eq. (8). The present iteration, which is between 1 and the complete number of iterations (M $Iter$), is denoted by C $Iter$. Min and Max represent the accelerated function's minimum and maximum values.

2.2.3. Exploration phase

The exploratory activity of AOA is presented in this section. Mathematical calculations using either the Division (D) operator or even the Multiplication (M) operator, according to the Arithmetic operators, have highly distributed values or decisions (refer to as separate reigns) dedicated to the process of exploration quest. However, because of their high dispersion, unlike other operators (S and A), these operators (D and M) cannot easily approach the goal. The discovery hunt then finds the near-optimal solution that can be inferred after many efforts (iterations). Furthermore, at this optimization stage, the discovery operators (D and M) were worked to assist the other stage (exploitation) in the search process by improved coordination between them.

In order to find the best solution based on two primary search strategies (Division (D) search strategy and Multiplication search strategy) modeled in Eq. (9), AOA's exploration operators explore the search field randomly in many areas. This search process (exploration search by running D or M, see Figure ??) is controlled by an accelerated Math Optimizer (MOA) function for a $r1$ >MOA condition ($r1$ is a random number). In this step (first factor in Eq. (9)), the first operator (D) is controlled by $r2$ <0.5 and the other operator (M) is ignored before the operator completes its current mission. Otherwise, instead of the D ($r2$ is a random number), the second operator (M) would be engaged to execute the current mission.

Notice, to create further diversification courses and explore different regions of the search space, a stochastic scaling coefficient is considered for the element. We used the simplified rule capable of simulating Arithmetic operators' behaviors. The following position-updating formulations for the exploration modules are presented.

$$x_{i,j}(C\_Iter + 1) = \begin{cases} best(x_j) \div (MOP + \epsilon) \times ((UB_j - LB_j) \times \mu + LB_j), & r2 < 0.5 \\ best(x_j) \times MOP \times ((UB_j - LB_j) \times \mu + LB_j), & otherwise \end{cases} \quad (9)$$

Where xi(C Iter+1) for the next iteration denotes the ith solution, xi, j(C Iter) in the next iteration represents the jth location of the ith solution in the current state, and $best(xj)$ in the best-obtained solution so far has been the jth position. ϵ is a small integer value, UBj and LBj represent the upper bound values and the lower bound values of the location of jth. μ is the control function used to change the search procedure, which is set to 0.5.

$$MOP(C\_Iter) = 1 - \frac{C\_Iter^{1/\alpha}}{M\_Iter^{1/\alpha}} \quad (10)$$

Where the probability of the Math Optimizer (MOP) is a parameter, MOP(C Iter) represents the tth iteration function value, and C $Iter$ denotes the current iteration, and (M $Iter$) indicates the cumulative number of iterations. α is a dynamic parameter that determines the precision of exploitation throughout iterations, set at 5.

2.2.4. Exploitation phase

AOA's exploitation strategy is presented in this section. The mathematical calculations either use the Subtraction (S) or Addition (A) have produced high-dense data, as per the Arithmetic operators, which apply to the exploitation search process. Conversely, unlike other operators, these operators (S and A) can easily reach the target because of their low dispersion, as seen in Figure ??. The exploitation process then finds the near-optimal solution that can be inferred after many efforts (iterations). In particular, at this optimization phase, the exploitation operators (S and A) worked to assist the exploitation process by improved coordination between them.

The MOA expected value for the $r1$ condition is not greater than the current MOA(C Iter) value (see Eq. (8)). The MOA function value controls this search stage (exploitation investigation by executing S or A) for the $r1$ status. In AOA, AOA's Subtraction (S) and Addition (A) operators thoroughly investigate the search area in many dense regions and approach to seeking a better solution based on two key research techniques (i.e., Subtraction (S) and Addition (A) search strategy) modeled in Eq. (11).

$$x_{i,j}(C\_Iter+1) = \begin{cases} best(x_j) - MOP \times ((UB_j - LB_j) \times \mu + LB_j), & r3 < 0.5 \\ best(x_j) + MOP \times ((UB_j - LB_j) \times \mu + LB_j), & otherwise \end{cases} \tag{11}$$

This stage takes advantage of the search space by doing an in-depth search, which appears very clearly in Figure ??. In this step (first law in Eq. (11)), the first operator (S) is controlled by $r3$ <0.5, and the other operator (A) is ignored before the operator completes its current mission. Otherwise, rather than S, the second operator (A) is employed to execute the current mission. These processes are identical to the partitions of the previous step in this stage. However, search operators (S and A) also strive to avoid being trapped in the local search field. This procedure aims to find the best solution for discovering search methods and preserving the variety of candidate solutions. The parameter μ is carefully constructed to create a stochastic value for each iteration to retain experimentation between the first and last iterations. This portion of searching is beneficial in local minima stagnation, especially over the last iterations.

To recap, the optimization in AOA starts with creating a randomized number of candidate solutions (population). During the repetition path, D, M, S, and A approximate the near-optimal solution's feasible positions. From the best-obtained solution, each solution revives its place. The MOA parameter is linearly improved from 0.2 to 0.9 to highlight discovery and exploitation. As $r1 > MOA$, evolutionary algorithms seek to diverge from the near-optimal solution and converge as $r1 < MOA$ toward the near-optimal solution. Finally, after obtaining the fulfillment of the end criteria, the AOA algorithm is halted. In Algorithm 1, the Pseudo-code of the conventional AOA is given.

Algorithm 1 Pseudo-code of the conventional AOA

```
Initialize the Arithmetic Optimization Algorithm parameters α, μ.
Initialize the solutions' positions randomly. (Solutions: i=1, ..., N.)
while (C_Iter < M_Iter) do
  Calculate the Fitness Function (FF) for the given solutions
  Find the best solution (Determined best so far).
  Update the MOA and MOP value using Equations (8) and (10).
  for (i=1 to Solutions) do
    for (j=1 to Positions) do
      Generate a random values between [0, 1] (r1, r2, and r3)
      if r1 >MOA then
        Exploration phase
        if r2 >0.5 then
          (1) Apply the Division math operator (D) using the first rule in Eq. (9).
        else
          (2) Apply the Multiplication math operator (M) using the second rule in Eq. (9).
        end if
      else
        Exploitation phase
        if r3 >0.5 then
          (1) Apply the Subtraction math operator (S) using the first rule in Eq. (11).
        else
          (2) Apply the Addition math operator (A) using the second rule in Eq. (11).
        end if
      end if
    end for
  end for
  C_Iter=C_Iter+1
end while
Return the best solution (x).
```

2.3. Integrated search strategies

This section presents the integrated search strategies to the proposed method as follows.

2.3.1. Opposition-based learning (OBL)

Opposition-based learning (OBL) [46] is a modern artificial intelligence technique established to be an essential idea to enhance multiple optimization methods [68]. The OBL strategy is taken from the existing approach to creating a new opposition solution for the given problem. This technique attempts to determine a better candidate solution by obtaining a better fitness score to approach the optimal solution [69]. The opposite value of X for the real value, where $X \in [UB, LB]$, is calculated using Eq. (12).

$$X = UB + LB - X \tag{12}$$

Opposite point [70]: let $X = (x-1, x-2, \ldots, x-n)$ be a multi-dimensional space point, where $x-1, x-2, \ldots, x-Dim \in R$ and $x-j\ [UB-j, LB-j], j \in 1, 2, \ldots, Dim$. This representation is used by adding Eq. (13) to resolve n-dimensions.

$$\overrightarrow{x_j} = UB_j + LB_j - x_j, \qquad where \quad j = 1 \ldots D. \tag{13}$$

In addition, the two solutions are given (x and x_{old}) and compared in the optimization procedure according to their fitness functions; the best solution is stored, and the other solution is removed. If $f(x) \geq f(x_{old})$ is saved for maximization, then x is saved; otherwise, x_{old} is saved.

2.3.2. Levy flight distribution (LFD)

Levy flight is one of the most efficient random distribution methods dependent on non-Gaussian distribution [71]. Eq. (14) is utilized to change the solution's position based on the Levy flight phase.

$$x(C\_Iter+1) = best(x_j) \times Levy(Dim) \tag{14}$$

$$Levy(D) = s \times \frac{u \times \sigma}{|\upsilon|^{\frac{1}{\beta}}} \tag{15}$$

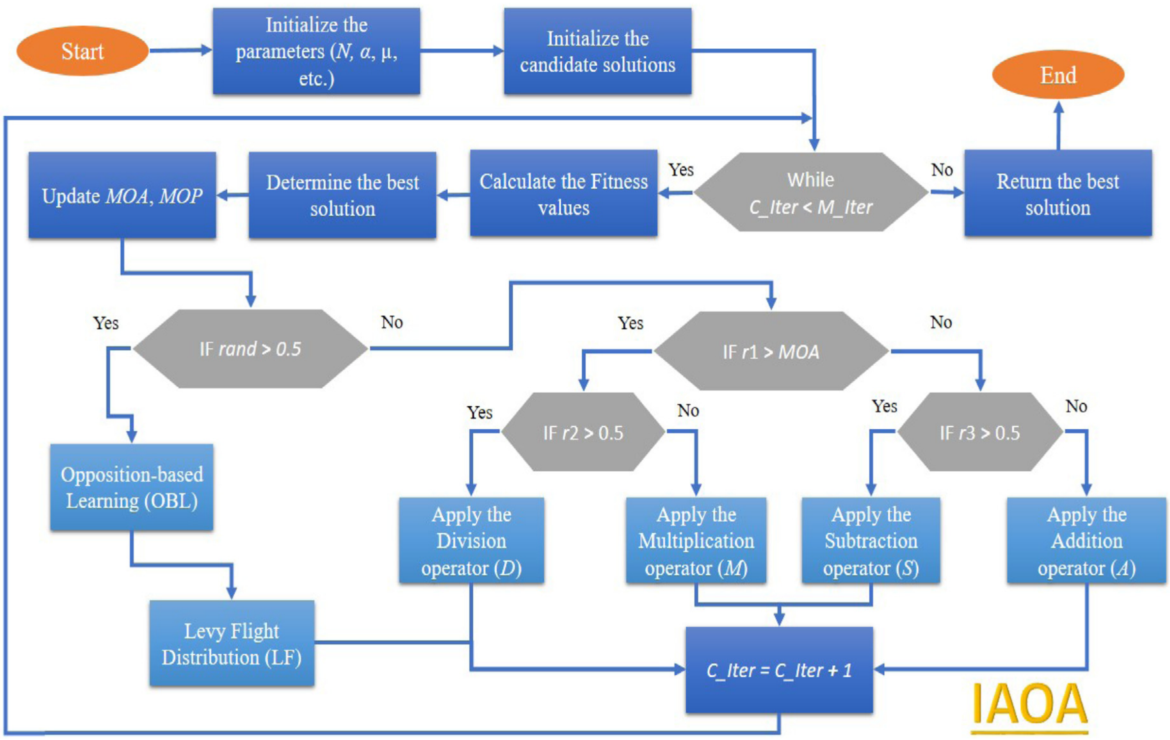

Fig. 2. Flowchart of the proposed improved Arithmetic Optimization Algorithm (IAOA).

where s is a constant value fixed to 0.01, u and υ are random numbers between 0 and 1. σ is calculated using Eq. (16).

$$\sigma = \left(\frac{\Gamma(1+\beta) \times sine(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}} \right) \tag{16}$$

where *sine* denotes the sine function value, and β is a constant value fixed to 1.5.

3. The proposed Improved Arithmetic Optimization Algorithm (IAOA)

In this section, we introduce a novel optimization method called IAOA to solve different text document clustering and benchmark optimization problems, as shown in Fig. 2. The proposed method consists of the conventional Arithmetic Optimization Algorithm (AOA), opposition-based learning (OBL), and the Levy flight distribution method.

The suggested IAOA is intended to alleviate the traditional approach's key limitations, which converge quickly/slowly and fall into the local optima trap when grappling with complicated and high dimensional problems. In the proposed IAOA method, two search techniques are applied. Firstly, opposition-based learning (OBL) is used to increase the AOA's convergence rate. Secondly, during the early and later stages, the Levy flight distribution (LFD) is used to optimize discovery and extraction searches, respectively. The two combined search approaches significantly increase the integration activity and the traditional AOA's searchability.

One primary condition in the proposed method is to execute the proposed processes after producing the candidate solutions randomly. This criterion applies the traditional AOA or the other two search techniques (i.e., OBL and Levy flight). If *rand* >0.5, the OBL and Levy flight are executed; if not, the AOA is executed. The intuitive and detailed process of the proposed IAOA is shown in Algorithm 2.

3.1. Complexity of IAOA

The developed IAOA depends on the complexity of traditional AOA, SCA, Levy flight, and OBL, and it is given as follows.

$$O(IAOA) = (N-K) \times O(AOA) + K \times (O(LF) + O(OBL)) \tag{17}$$

$$O(AOA) = O(N \times (t \times Dim + 1)) \tag{18}$$

$$O(LF) = O(N \times Dim) \tag{19}$$

$$O(OBL) = O(N \times Dim) \tag{20}$$

Therefore, the complexity of the developed IAOA is given as

$$O(IAOA) = O\,(t \times ((N-K) \times N \times (Dim+1)) \\ + K\,(N \times Dim + N \times Dim)) \tag{21}$$

$$O(IAOA) = O\,(t \times ((N-K) \times N \times (Dim+1)) + K\,(N \times Dim)) \tag{22}$$

$$O(IAOA) = O\,(t \times N \times (Dim + N - K)) \tag{23}$$

4. Experiments results and discussion

In this section, we evaluate the performance of the developed IAOA by using two different applications. In the first application, the IAOA is applied to solve a set of CEC2019 optimization problems. Whereas, the second one aims to assess the

Algorithm 2 Pseudo-code of the IAOA algorithm

```
Initialize the Arithmetic Optimization Algorithm parameters α, μ.
Initialize the solutions' positions randomly (X). (Solutions: i=1, ..., N.)
while (C_Iter < M_Iter) do
  Calculate the Fitness Function (FF) for the given solutions
  Find the best solution (Determined best so far).
  Update the MOA value using Eq. (8).
  Update the MOP value using Eq. (10).
  if rand >0.5 then
    Apply the OBL ∀ X
    Apply the LF ∀ X
  else
    for (i=1 to Solutions) do
      for (j=1 to Positions) do
        Generate a random values between [0, 1] (r1, r2, and r3)
        if r1 >MOA then
          Exploration phase
          if r2 >0.5 then
            (1) Apply the Division math operator (D " ÷ ").
            Update the i_th solutions' positions using the first rule in Eq. (9).
          else
            (2) Apply the Multiplication math operator (M " × ").
            Update the i_th solutions' positions using the second rule in Eq. (9).
          end if
        else
          Exploitation phase
          if r3 >0.5 then
            (1) Apply the Subtraction math operator (S " − ").
            Update the i_th solutions' positions using the first rule in Eq. (11).
          else
            (2) Apply the Addition math operator (A " + ").
            Update the i_th solutions' positions using the second rule in Eq. (11).
          end if
        end if
      end for
    end for
  end if
  C_Iter=C_Iter+1
end while
Return the best solution (x).
```

applicability of IAOA. In addition, the results of the developed IAOA are compared with other methods including gray-wolf optimization (GWO) [72], Harris Hawks Optimizer (HHO) [73], faramarzi2020equilibrium (EO) [74], Marine Predators algorithm (MPA) [75,76], multi-verse optimization (MVO) [77], Particle swarm optimization (PSO) [78], Sine–cosine algorithm (SCA) [79], salp swarm algorithm (SSA) [80,81], whale optimization algorithm (WOA) [82], and the traditional AOA.

Table 1 lists the parameter settings for each algorithm. All of the methods are executed by the MATLAB 2015a program. A 16 GB RAM Intel Core(TM) i7 1.80 GHz 2.30 GHz processor is used to perform these algorithms. The number of solutions that can be employed is limited to 30. For a fair comparison, the maximum number of iterations is set to 500. Each competitor algorithm generates thirteen independent runs.

4.1. Experiments 1: CEC2019 benchmark functions

This section will study the modified IAOA algorithm's performance to solve a set of benchmark functions from CEC2019. The description of these functions is given in Table 2, which illustrates the optimal solution, dimension, and the search space of the ten functions.

This evaluation compared the proposed IAOA with a set of nine MH techniques and the traditional AOA. The parameters of each algorithm are set according to the original implementation as given before. Table 3 and Fig. 3 show the results obtained by the implemented algorithm. It can be noticed from Table 3 that the developed IAOA has the best rank at four functions from ten, which represents 40% of the total functions. The IAOA has also achieved the second rank in four functions (CEC02, CEC04, CEC06, and CEC09), whereas it has the fourth and third rank for the CEC05 and CEC07, respectively. The aforementioned indicates the high ability of IAOA to find the optimal solution inside the search space of the complex CEC2019 functions.

Fig. 3 depicts the convergence of the algorithms overall the tested functions. By inspecting the figure, it can be seen that the IAOA has a high ability to converge to the optimal solution compared to the other methods. The classical AOA shows high stagnation property toward the local solutions in most functions. Through these observations, the IAOA affirms its high ability to leave the local solutions and discover the search space efficiently compared to the AOA, proving the remarkable impact of merging the OBL and LF operators in the classical AOA.

Overall, this section's results show the benefits of Opposition-based learning (OBL) and Levy Flight distribution (LFD) in enhancing conventional AOA performance using CEC2019 problems. The proposed IAOA obtained better results in almost all the test cases and promising results compared with other well-known algorithms. The more precise results of IAOA are due to the best exploitation of AOA and OBL and the super-exploitation of LFD. For AOA to be very precise, the incorporated methods are performed according to dynamic exchange strategies as given in Section 3. This paper revealed that the proposed IAOA is a beneficial method due to the proactive characteristics of the best-obtained solutions by the MPA. The trustworthiness and robustness of the IAOA are from the high exploration and local optima escape by utilizing the LFD search method. This effective control scheme of the exploration and exploitation methods allowed IAOA to achieve more reliable outcomes than the other algorithms.

4.1.1. Comparisons with the state-of-the-art methods

In this section, the proposed IAOA is further evaluated using ten CEC2019 and compared to the state-of-the-art methods published in the literature. The comparative methods include Fuzzy Self-Tuning PSO (FST-PSO) [84], improved BA with variable neighborhood (VNBA) [85], novel PSO using prey–predator relationship (PP-PSO) [86], Hybrid KHA with differential evolution (DEKH) [87], Chaotic CS (CCS) [88], and stud krill herd algorithm (SKH) [89].

Table 4 shows the results obtained by the proposed IAOA and the best-published results using CEC2019. The average values of the comparative methods are given, and it is clear that the proposed IAOA is a promising search method according to its obtained results. The proposed IAOA method results are excellent compared to the other modified state-of-the-art methods taken from previous studies. This means that the proposed IAOA can solve complex problems such as this type of problem (CEC2019).

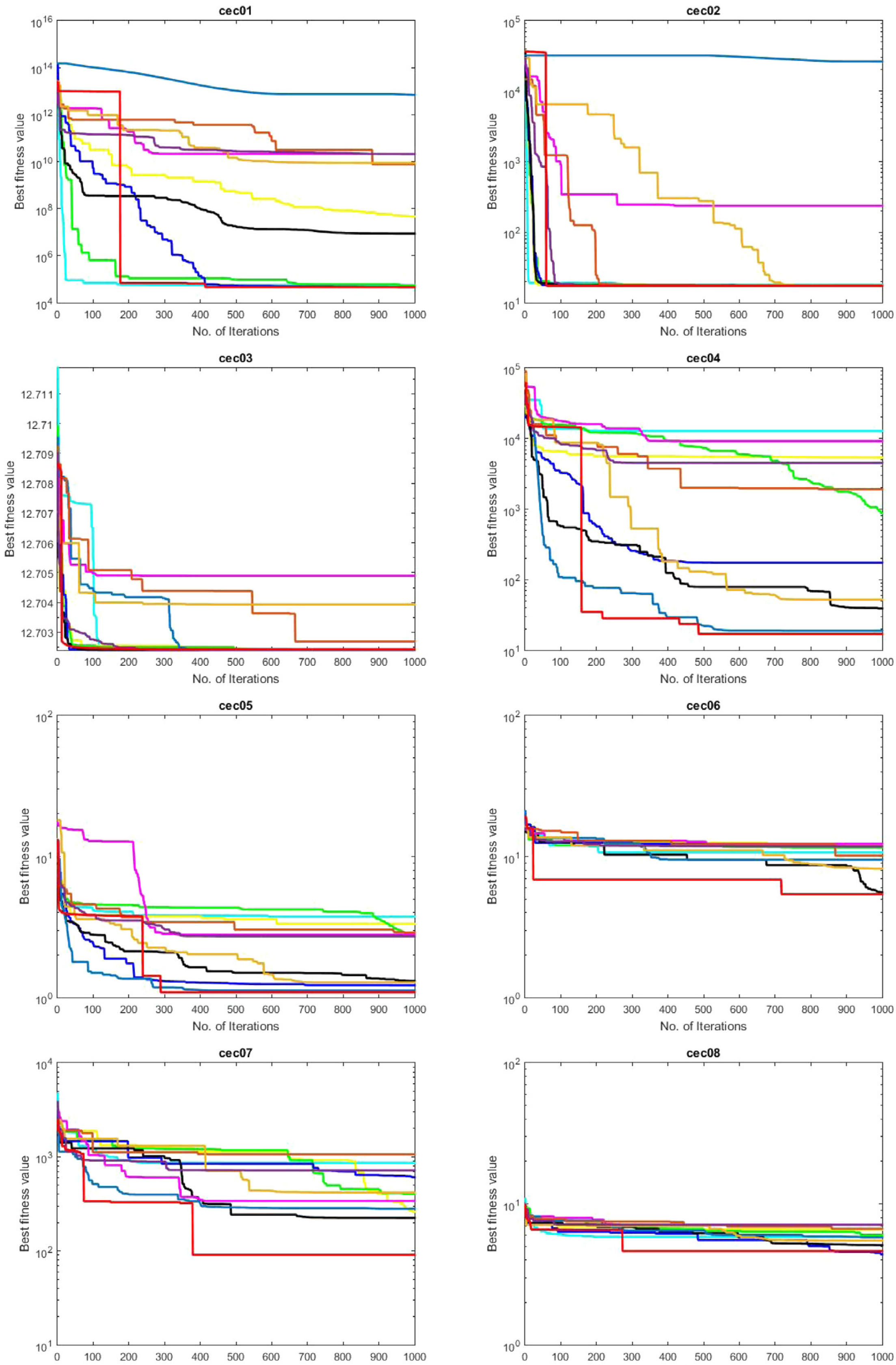

Fig. 3. Convergence behavior of the comparative methods on the tested benchmark functions (CEC2019).

Table 1
Parameters setting of the comparative methods.

No.	Algorithm	Reference	Parameter	Value
1	AOA	[45]	α	5
			μ	0.5
2	GWO	[72]	Convergence parameter (a)	Linear reduction from 2 to 0
3	HHO	[73]	α	1.5
4	EO	[83]	r	0.5
			a	4
			GP	0.5
5	MPA	[76]	γ	$\gamma > 1$
			P	0.0
6	MVO	[77]	p	6
7	PSO	[78]	Topology	Fully connected
			Cognitive and social constant	(C1, C2) 2, 2
			Inertia weight	Linear reduction from 0.9 to 0.1
			Velocity limit	10% of dimension range
8	SCA	[79]	α	0.05
9	SSA	[81]	v_0	0
10	WOA	[82]	α	Decreased from 2 to 0
			b	2

Table 2
Review of CEC2019 benchmark function problems.

No.	Functions	$F_i^* = F_i(x^*)$	Dim	Search range
1	Storn's Chebyshev Polynomial Fitting Problem	1	9	[−8192, 8192]
2	Inverse Hilbert Matrix Problem	1	16	[−16384, 16384]
3	Lennard-Jones Minimum Energy Cluster	1	18	[−4, 4]
4	Rastrigin's Function	1	10	[−100, 100]
5	Griewangk's Function	1	10	[−100, 100]
6	Weierstrass Function	1	10	[−100, 100]
7	Modified Schwefel's Function	1	10	[−100, 100]
8	Expanded Schaffer's F6 Function	1	10	[−100, 100]
9	Happy Cat Function	1	10	[−100, 100]
10	Ackley Function	1	10	[−100, 100]

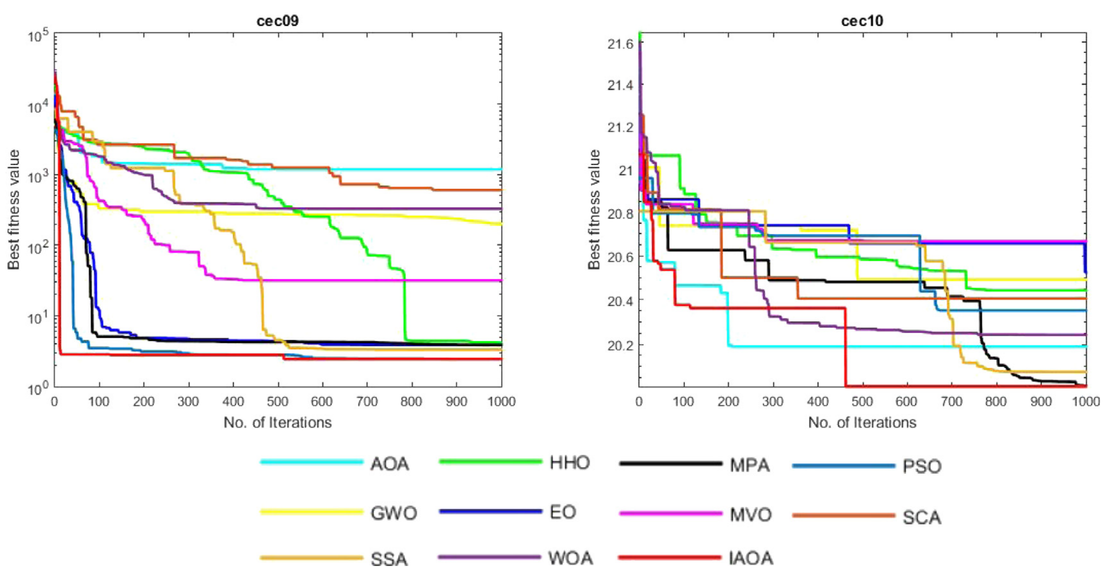

Fig. 3. (*continued*).

The evaluation method using the Friedman ranking test indicates that the proposed IAOA achieved the best results and was ranked first among all the table's comparison methods. Fig. 4 shows the final ranking of the comparative methods using the Friedman ranking test. VNBA got the second-ranking, followed by DEKH, PP-PSO, SKH, and CCS.

4.2. Experiments 2: Text clustering problem

In this section, the applicability of the IAOA in a real-world application of text document clustering is assessed. The utilized datasets, the performed statistical analyzes, and discussions are explained in the following subsections.

4.2.1. Benchmark datasets

In this section, ten various benchmark datasets are used to validate the performance of the proposed IAOA, as shown in Table 5. These datasets are taken from Laboratory of Computational Intelligence (LABIC), which belong to three common benchmark datasets (i.e., Dmoz-Health-spars, Reuters-21578, and 20News-group) with different sizes of documents, terms, and clusters. The Dmoz-Health-sparse dataset holds several health care topics, including public health and safety, conditions and diseases, animal,

Table 3
The results of the comparative methods using advanced CEC2019 benchmark functions.

Function	Measure	Comparative algorithms										
		AOA	GWO	HHO	EO	MPA	MVO	PSO	SCA	SSA	WOA	IAOA
CEC01	Worst	2.6576E+11	3.1721E+09	1.3186E+05	3.6634E+08	5.0891E+08	3.1610E+11	8.5146E+12	1.6668E+11	8.4762E+10	3.4414E+11	5.3135E+04
	Average	1.7737E+11	1.1738E+09	9.4791E+04	1.5620E+08	2.3783E+08	1.7595E+11	6.7091E+12	6.2864E+10	4.3463E+10	1.7942E+11	5.0009E+04
	Best	2.8538E+10	7.2262E+05	5.7776E+04	1.0356E+06	3.4724E+06	5.2680E+10	4.5129E+12	2.1879E+09	1.0158E+10	4.8572E+09	4.7652E+04
	STD	1.2965E+11	1.7393E+09	3.7042E+04	1.8875E+08	2.5471E+08	1.3252E+11	2.0293E+12	9.0337E+10	3.7939E+10	1.6985E+11	2.8218E+03
	P-value	7.6854E−02	3.0733E−01	1.1402E−02	2.2506E−01	1.8114E−01	8.2961E−02	4.6038E−03	2.9451E−01	1.1823E−01	1.4130E−01	6.7104E−06
	h	0	0	1	0	0	0	1	0	0	0	NAN
	Rank	9	5	2	3	4	8	11	7	6	10	1
CEC02	Worst	1.8559E+01	1.7345E+01	1.7400E+01	1.7346E+01	1.7343E+01	4.8638E+03	3.1718E+04	1.7808E+01	1.7624E+01	1.8272E+01	1.7344E+01
	Average	1.8168E+01	1.7345E+01	1.7380E+01	1.7345E+01	1.7343E+01	2.5122E+03	2.8713E+04	1.7770E+01	1.7483E+01	1.7900E+01	1.7344E+01
	Best	1.7923E+01	1.7344E+01	1.7366E+01	1.7343E+01	1.7343E+01	1.2208E+03	2.6769E+04	1.7747E+01	1.7353E+01	1.7361E+01	1.7343E+01
	STD	3.4231E−01	4.3922E−04	1.7997E−02	1.6649E−03	2.7298E−06	2.0398E+03	2.6402E+03	3.2910E−02	1.3580E−01	4.7796E−01	5.0520E−04
	P-value	1.4053E−02	1.5671E−01	2.6033E−02	3.2173E−01	2.3371E−02	1.0152E−01	4.6888E−05	2.3455E−05	1.5131E−01	1.1392E−01	1.0000E+00
	h	1	0	1	0	1	0	1	1	0	0	NAN
	Rank	9	3	5	4	1	10	11	7	6	8	2
CEC03	Worst	1.2702E+01	1.2702E+01	1.2702E+01	1.2702E+01	1.2702E+01	1.2705E+01	1.2702E+01	1.2704E+01	1.2702E+01	1.2703E+01	1.2702E+01
	Average	1.2702E+01	1.2702E+01	1.2702E+01	1.2702E+01	1.2702E+01	1.2703E+01	1.2702E+01	1.2703E+01	1.2702E+01	1.2702E+01	1.2702E+01
	Best	1.2702E+01	1.2702E+01	1.2702E+01	1.2702E+01	1.2702E+01	1.2702E+01	1.2702E+01	1.2702E+01	1.2702E+01	1.2702E+01	1.2702E+01
	STD	8.9826E−06	3.3413E−05	1.7883E−05	4.5377E−09	6.9747E−10	1.3070E−03	1.6009E−08	7.8204E−04	1.6179E−08	9.2358E−05	1.8253E−03
	P-value	3.3793E−01	3.3889E−01	3.4021E−01	3.3145E−01	3.3145E−01	9.3099E−01	3.3145E−01	6.7051E−01	3.3145E−01	3.6467E−01	1.0000E+00
	h	0	0	0	0	0	0	0	0	0	0	NAN
	Rank	6	7	8	4	2	11	3	10	5	9	1
CEC04	Worst	1.3203E+04	2.4810E+03	3.0529E+03	1.5998E+03	7.5969E+01	3.7961E+04	5.1678E+01	3.0464E+03	1.3348E+02	8.2500E+03	6.4818E+01
	Average	1.1832E+04	8.7147E+02	2.5488E+03	6.8850E+02	6.5721E+01	1.8817E+04	2.9641E+01	2.0497E+03	7.4834E+01	6.4007E+03	5.2217E+01
	Best	9.5669E+03	6.3311E+01	1.7988E+03	1.0191E+02	5.6972E+01	6.8603E+03	1.7236E+01	1.4582E+03	4.4263E+01	5.3275E+03	3.9131E+01
	STD	1.9762E+03	1.3939E+03	6.6223E+02	7.9999E+02	9.5866E+00	1.6750E+04	1.9135E+01	8.6813E+02	5.0801E+01	1.6084E+03	1.2850E+01
	P-value	4.9661E−04	3.6627E−01	2.8433E−03	2.4043E−01	2.1836E−01	1.2433E−01	1.6503E−01	1.6333E−02	4.9627E−01	2.3951E−03	1.0000E+00
	h	1	0	1	0	0	0	0	1	0	1	NAN
	Rank	10	6	8	5	3	11	1	7	4	9	2
CEC05	Worst	5.2521E+00	1.9438E+00	3.1866E+00	1.6628E+00	1.4025E+00	3.4094E+00	1.4747E+00	2.4433E+00	1.3043E+00	2.9187E+00	1.6561E+00
	Average	4.6512E+00	1.6263E+00	2.6416E+00	1.4340E+00	1.3462E+00	2.8887E+00	1.2683E+00	2.4397E+00	1.2027E+00	2.7407E+00	1.4226E+00
	Best	4.2751E+00	1.2340E+00	2.1114E+00	1.2092E+00	1.2460E+00	2.1646E+00	1.1245E+00	2.4336E+00	1.1017E+00	2.6296E+00	1.3018E+00
	STD	5.2590E−01	3.6074E−01	5.3777E−01	2.2680E−01	8.7014E−02	6.4689E−01	1.8328E−01	5.2883E−03	1.0134E−01	1.5577E−01	2.0223E−01
	P-value	5.7865E−04	4.4168E−01	2.1296E−02	9.5125E−01	5.8043E−01	2.0002E−02	3.8303E−01	9.5763E−04	1.6750E−01	8.6449E−04	1.0000E+00
	h	1	0	1	0	0	1	0	1	0	1	NAN
	Rank	11	6	8	5	3	10	2	7	1	9	4
CEC06	Worst	1.1517E+01	1.2633E+01	1.2735E+01	1.2513E+01	7.8589E+00	1.2485E+01	1.0776E+01	1.2907E+01	9.1684E+00	1.0072E+01	7.6067E+00
	Average	9.8818E+00	1.2132E+01	1.1781E+01	1.1744E+01	6.4673E+00	1.1754E+01	1.0142E+01	1.1952E+01	8.4898E+00	9.7219E+00	7.0579E+00
	Best	8.9563E+00	1.1218E+01	1.0930E+01	1.0956E+01	4.9277E+00	1.0518E+01	9.8024E+00	1.0777E+01	8.0564E+00	9.0299E+00	6.4619E+00
	STD	1.4200E+00	7.9264E−01	9.0647E−01	7.7855E−01	1.4712E+00	1.0759E+00	5.4897E−01	1.0817E+00	5.9519E−01	5.9938E−01	5.7386E−01
	P-value	3.3101E−02	8.5050E−04	1.5877E−03	1.1033E−03	5.5241E−01	2.6253E−03	2.5436E−03	2.2857E−03	3.9951E−02	5.1211E−03	1.0000E+00
	h	1	1	1	1	0	1	1	1	1	1	NAN
	Rank	5	11	9	7	1	8	6	10	3	4	2
CEC07	Worst	1.1023E+03	1.2010E+03	7.3496E+02	5.5304E+02	3.0693E+02	1.3714E+03	4.2574E+02	1.1820E+03	7.8592E+02	7.3027E+02	3.9566E+02
	Average	8.1001E+02	8.7062E+02	4.2720E+02	4.5434E+02	1.6522E+02	8.2994E+02	2.2334E+02	1.0641E+03	5.4810E+02	5.0106E+02	2.4840E+02
	Best	5.9248E+02	2.5511E+02	1.2502E+02	3.4763E+02	2.7866E+01	5.2694E+02	8.6866E+01	9.4210E+02	2.0212E+02	2.9477E+02	9.7065E+01
	STD	2.6298E+02	5.3352E+02	3.0501E+02	1.0294E+02	1.3958E+02	4.7002E+02	1.7880E+02	1.2000E+02	3.0656E+02	2.1865E+02	1.4934E+02
	P-value	3.2389E−02	1.2363E−01	4.1340E−01	1.2064E−01	5.1981E−01	1.1065E−01	8.6127E−01	1.8018E−03	2.0260E−01	1.7373E−01	1.0000E+00
	h	1	0	0	0	0	0	0	1	0	0	NAN
	Rank	8	10	4	5	1	9	2	11	7	6	3
CEC08	Worst	6.3410E+00	6.8419E+00	7.0641E+00	5.6025E+00	5.7696E+00	6.9799E+00	6.4475E+00	7.3441E+00	6.3053E+00	6.5569E+00	4.9512E+00
	Average	5.8486E+00	6.1242E+00	6.5066E+00	5.2538E+00	5.5647E+00	6.6919E+00	6.0802E+00	7.1469E+00	5.8450E+00	6.1804E+00	4.3291E+00
	Best	5.5362E+00	4.8221E+00	6.1889E+00	5.0053E+00	5.1708E+00	6.1736E+00	5.6623E+00	6.8066E+00	5.0275E+00	5.6622E+00	3.6455E+00
	STD	4.3151E−01	1.1296E+00	4.8437E−01	3.1099E−01	3.4124E−01	4.4979E−01	3.9507E−01	2.9596E−01	7.0985E−01	4.6393E−01	6.5500E−01
	P-value	2.8425E−02	7.5904E−02	9.8096E−03	9.1749E−02	4.4222E−02	6.7422E−03	1.6608E−02	2.4565E−03	5.3076E−02	1.6198E−02	1.0000E+00
	h	1	0	1	0	1	1	1	1	0	1	NAN
	Rank	5	7	9	2	3	10	6	11	4	8	1
CEC09	Worst	1.4282E+03	2.9719E+02	4.0339E+01	5.8417E+00	5.8447E+00	3.1811E+02	2.6676E+00	6.3071E+02	5.5221E+00	1.4863E+03	3.3948E+00
	Average	1.2255E+03	1.0249E+02	2.5294E+01	4.8183E+00	4.4008E+00	2.0424E+02	2.5543E+00	4.2613E+02	4.6994E+00	8.1876E+02	3.1752E+00
	Best	9.4740E+02	4.8061E+00	5.6035E+00	3.9101E+00	3.6093E+00	2.2935E+01	2.4525E+00	2.1791E+02	3.7386E+00	2.9562E+02	2.8311E+00
	STD	2.4911E+02	1.6861E+02	1.7828E+01	9.7092E−01	1.2524E+00	1.5872E+02	1.0798E−01	2.0642E+02	8.9977E−01	6.0835E+02	3.0179E−01
	P-value	1.0512E−03	3.6530E−01	9.8132E−02	4.8858E−02	1.7474E−01	9.3262E−02	2.8431E−02	2.3821E−02	4.9726E−02	8.0953E−02	1.0000E+00
	h	1	0	0	1	0	0	1	1	1	0	NAN
	Rank	11	7	6	5	3	8	1	9	4	10	2
CEC10	Worst	2.0722E+01	2.0480E+01	2.0525E+01	2.0662E+01	2.0718E+01	2.0681E+01	2.0801E+01	2.0607E+01	2.0502E+01	2.0511E+01	2.0596E+01
	Average	2.0566E+01	2.0382E+01	2.0148E+01	2.0441E+01	2.0663E+01	2.0568E+01	2.0621E+01	2.0405E+01	2.0213E+01	2.0339E+01	1.9945E+01
	Best	2.0426E+01	2.0316E+01	2.0000E+01	2.0291E+01	2.0534E+01	2.0368E+01	2.0353E+01	2.0050E+01	2.0000E+01	2.0209E+01	1.8019E+01
	STD	1.2382E−01	7.6077E−02	2.3045E−01	1.4270E−01	7.4195E−02	1.5118E−01	1.6692E−01	2.3151E−01	2.4422E−01	1.4867E−01	1.0842E+00
	P-value	3.0855E−02	5.8423E−01	1.5709E−01	3.0197E−01	2.4327E−03	4.2325E−02	2.2661E−02	6.0948E−01	3.5203E−01	3.0847E−01	4.4352E−01
	h	1	0	0	0	1	1	1	0	0	0	NAN
	Rank	8	5	2	7	11	9	10	6	3	4	1
Summation		82	67	61	47	32	94	53	85	43	77	19
Mean rank		8.20	6.70	6.10	4.70	3.20	9.40	5.30	8.50	4.30	7.70	1.90
Final Ranking		9	7	6	4	2	11	5	10	3	8	1

mental health, nutrition, pharmacy, addictions, nursing, medicine, and others. Reuters-21578 dataset holds several news and topics: gold, tin, hog, citrus pulp, gas, peseta, housing, cotton, lumber, interest, inventories, linseed, rand, rice, tapioca, and others. 20Newsgroup dataset holds several news and topics: computer windows x, talk politics misc, computer system mac hardware, talk religion misc, science electronics, computer graphics, talk politic mid-east, social religion Christian, etc. As shown in Table 5, dataset 1 (i.e., DAS1) contains 40 documents, 301 terms, and 2 clusters. Dataset 2 (i.e., DAS2) contains 80 documents, 471 terms, and 2 clusters. Dataset 3 (i.e., DAS3) contains 120 documents, 657 terms, and 3 clusters. Dataset 4 (i.e., DAS4) contains 160 documents, 807 terms, and 4 clusters. Dataset 5 (i.e., DAS5) contains 200 documents, 983 terms, and 5 clusters. Dataset 6 (i.e., DAS6) contains 240 documents, 1095 terms, and 6 clusters. Dataset 7 (i.e., DAS7) contains 280 documents, 1157 terms, and 7 clusters. Dataset 8 (i.e., DAS8) contains 320 documents, 1282 terms, and 8 clusters. Dataset 9 (i.e., DAS9) contains 360 documents, 1405 terms, and 9 clusters. Finally, dataset 10 (i.e., DAS10) contains 400 documents, 1502 terms, and 10 clusters.

4.2.2. Evaluation measures

In this section, the performance measures used to assess the quality of the results obtained by each algorithm are given—for example, accuracy, recall, precision, F-measure, Purity, and Entropy.

1. Accuracy: This measure computes the ratio of the corrected predicted samples to the total number of samples, and it is formulated as follows:

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{24}$$

Table 4
The results obtained by the proposed IAOA and best-published results using CEC2019.

Dataset	Measure	Comparative algorithms						
		IAOA	PP-PSO	VNBA	FST-PSO	DEKH	CCS	SKH
CEC01	Average	5.0009E+04	2.3500E+08	2.5509E+06	8.8462E+06	9.7534E+05	2.1141E+05	1.7732E+06
	Rank	1	7	5	6	3	2	4
CEC02	Average	1.7344E+01	2.6635E+04	8.1681E+02	3.2288E+03	7.0806E+02	3.5892E+02	4.0272E+02
	Rank	1	7	5	6	4	2	3
CEC03	Average	1.2702E+01	6.6440E+00	4.6163E+00	9.5409E+00	6.5873E+00	1.2482E+01	8.0693E+00
	Rank	7	3	1	5	2	6	4
CEC04	Average	5.2217E+01	4.7666E+01	2.4728E+01	5.8252E+01	2.9409E+01	6.1460E+01	5.0955E+01
	Rank	5	3	1	6	2	7	4
CEC05	Average	1.4226E+00	1.6605E+00	3.0251E+00	2.0034E+01	1.9507E+00	3.4583E+00	1.9351E+00
	Rank	1	2	5	7	4	6	3
CEC06	Average	7.0579E+00	8.1572E+00	5.7135E+00	9.7719E+00	3.1953E+00	6.6010E+00	1.0254E+01
	Rank	4	5	2	6	1	3	7
CEC07	Average	2.4840E+02	1.1194E+03	6.4463E+02	1.3360E+03	1.0600E+03	2.2474E+03	1.3488E+03
	Rank	1	4	2	5	3	7	6
CEC08	Average	4.3291E+00	4.3978E+00	4.7793E+00	4.7774E+00	4.4510E+00	5.4003E+00	4.8795E+00
	Rank	1	2	5	4	3	7	6
CEC09	Average	3.1752E+00	1.5280E+00	1.3950E+00	1.6020E+00	3.3615E+00	1.6453E+00	1.5981E+00
	Rank	6	2	1	4	7	5	3
CEC10	Average	1.9945E+01	2.1380E+01	2.1080E+01	2.1147E+01	2.1638E+01	2.1960E+01	2.1602E+01
	Rank	1	4	2	3	6	7	5
Summation		28	39	29	52	35	52	45
Mean Ranking		3	4	3	5	4	5	5
Final Ranking		1	4	2	6	3	6	5

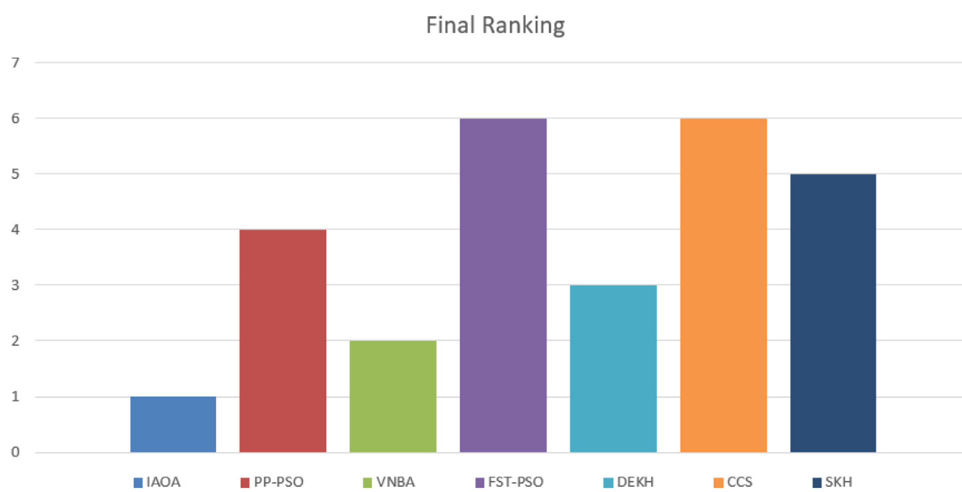

Fig. 4. Final results of the comparative state-of-the-art methods using Friedman ranking test.

Table 5
Datasets details.

Datasets	Source	Number of documents	Number of terms	Number of clusters
DAS1	Dmoz-Health-sparse	40	301	02
DAS2	Dmoz-Health-sparse	80	471	02
DAS3	Dmoz-Health-sparse	120	657	03
DAS4	Dmoz-Health-sparse	160	807	04
DAS5	Dmoz-Health-sparse	200	983	05
DAS6	Dmoz-Health-sparse	240	1095	06
DAS7	Dmoz-Health-sparse	280	1157	07
DAS8	Dmoz-Health-sparse	320	1282	08
DAS9	Reuters-21578	360	1405	09
DAS10	20Newsgroup	400	1502	10

where, *TP* denotes the true positive, *TN* denotes the true negative, *FP* denotes the false positive, and *FN* denotes the false negative.

2. Recall: it is defined, also as the true positive rate (TPR), which represents the percentage of predicting positive samples, and it is formulated as follows:

$$Recall = \frac{TP}{(TP + FN)} \tag{25}$$

3. Precision: It is defined as the ratio of the relevant samples among all retrieved samples.

$$Precision = \frac{TP}{(TP + FP)} \tag{26}$$

Table 6
The results of the comparative methods using dataset 1 (DAS1).

Measure	Metric	Comparative algorithms										
		AOA	GWO	HHO	EO	MPA	MVO	PSO	SCA	SSA	WOA	IAOA
	Best	0.6250	0.6500	0.6750	0.6500	0.6500	0.5750	0.1750	0.6500	0.5500	0.6500	0.6500
	Average	0.5875	0.5875	0.6250	0.6500	0.6500	0.5250	0.1500	0.6500	0.5375	0.6500	0.6500
Accuracy	Worst	0.5500	0.5250	0.5750	0.6500	0.6500	0.4750	0.1250	0.6500	0.5250	0.6500	0.6500
	Std	0.0530	0.0884	0.0707	0.0000	0.0000	0.0707	0.0354	0.0000	0.0177	0.0000	0.0000
	Rank	7	7	6	1	1	10	11	1	9	1	1
	Best	0.3846	0.3939	0.5806	0.3939	0.3939	0.5994	0.2817	0.3939	0.5069	0.3939	0.3939
	Average	0.3697	0.3691	0.5712	0.3939	0.3939	0.5063	0.2144	0.3939	0.4648	0.3939	0.3939
F-measure	Worst	0.3548	0.3443	0.5618	0.3939	0.3939	0.4133	0.1471	0.3939	0.4226	0.3939	0.3939
	Std	0.0211	0.0351	0.0133	0.0000	0.0000	0.1316	0.0952	0.0000	0.0596	0.0000	0.0000
	Rank	9	10	1	4	4	2	11	4	3	4	4
	Best	0.3421	0.1029	2.1422	0.0256	0.0256	0.7555	1.3000	0.0256	1.1500	0.3462	0.0256
	Average	0.3353	0.0643	1.4309	0.0256	0.0256	0.7426	0.8286	0.0256	0.9850	0.1859	0.0256
Purity	Worst	0.3286	0.0256	0.7197	0.0256	0.0256	0.7298	0.3571	0.0256	0.8200	0.0256	0.0256
	Std	0.0096	0.0547	1.0058	0.0000	0.0000	0.0182	0.6667	0.0000	0.2333	0.2266	0.0000
	Rank	5	7	1	8	8	4	3	8	2	6	8
	Best	0.2019	0.0821	0.3276	0.0821	0.0821	0.2500	0.0035	0.0821	0.0268	0.0821	0.0821
	Average	0.2042	0.1073	0.9571	0.0821	0.0821	0.2876	0.0943	0.0821	0.1403	0.1448	0.0821
Entropy	Worst	0.2064	0.1325	1.5867	0.0821	0.0821	0.3253	0.1851	0.0821	0.2539	0.2074	0.0821
	Std	0.0032	0.0357	0.8903	0.0000	0.0000	0.0533	0.1284	0.0000	0.1606	0.0886	0.0000
	Rank	9	6	11	1	1	10	5	1	7	8	1
	Best	0.3289	0.3333	0.6029	0.3333	0.3333	0.5934	0.8333	0.3333	0.5067	0.3333	0.3333
	Average	0.3216	0.3211	0.5805	0.3333	0.3333	0.5041	0.5952	0.3333	0.4617	0.3333	0.3333
Precision	Worst	0.3143	0.3088	0.5581	0.3333	0.3333	0.4148	0.3571	0.3333	0.4167	0.3333	0.3333
	Std	0.0104	0.0173	0.0317	0.0000	0.0000	0.1263	0.3367	0.0000	0.0636	0.0000	0.0000
	Rank	10	11	2	5	5	3	1	5	4	5	5
	Best	0.4630	0.4815	0.5655	0.4815	0.4815	0.6054	0.1695	0.4815	0.5071	0.4815	0.4815
	Average	0.4352	0.4352	0.5627	0.4815	0.4815	0.5085	0.1311	0.4815	0.4679	0.4815	0.4815
Recall	Worst	0.4074	0.3889	0.5598	0.4815	0.4815	0.4117	0.0926	0.4815	0.4288	0.4815	0.4815
	Std	0.0393	0.0655	0.0040	0.0000	0.0000	0.1370	0.0544	0.0000	0.0554	0.0000	0.0000
	Rank	9	9	1	3	3	2	11	3	8	3	3
	Best	0.4898	0.5551	0.3736	0.5502	0.5600	0.3609	0.4982	0.5524	0.4012	0.5180	0.5565
	Average	0.4648	0.5183	0.3646	0.5450	0.5472	0.3589	0.4715	0.5192	0.3971	0.5018	0.5515
Fitness	Worst	0.4399	0.4816	0.3556	0.5397	0.5344	0.3570	0.4447	0.4861	0.3930	0.4856	0.5464
	Std	0.0353	0.0519	0.0127	0.0074	0.0181	0.0027	0.0379	0.0468	0.0058	0.0229	0.0072
	Rank	8	5	10	3	2	11	7	4	9	6	1
Mean	Rank	8.14	7.86	4.57	3.57	3.43	6.00	7.00	3.71	6.00	4.71	3.29
Final	Ranking	11	10	5	3	2	7	9	4	7	6	1

4. F-measure: It is defined as the harmonic average of the recall and precision for each group.

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (27)$$

5. Entropy: It is applied to assess the information content of each cluster, and it also refers to the weighted average of each cluster, and it is formulated as follows:

$$Entropy = \sum_{i=1}^{K} \frac{m_i}{m} Ent_i, \;\; Ent_i = \sum_{j=1}^{L} p_{ij} log(p_{ij}), \;\; p_{ij} = \frac{m_{ij}}{m_j} \quad (28)$$

where K and L is the number of clusters and classes, respectively. m_{ij} and m_j are the number of objects that belong to the ith class in class j and is the number of objects that belong to the jth, respectively.

6. Purity: It computes the quality of the cluster consists of objects from the same class.

$$Purity = \sum_{i=1}^{K} \frac{m_i}{m} Pur_j, \quad Pur_j = max\, p_{ij}, \quad (29)$$

where m is the total number of objects.

4.2.3. Results and discussions

In this section, the proposed method is evaluated using nine datasets, as shown in Table 5. The results of these experiments are recorded in Tables 6–15. These tables contain seven measures: accuracy, f-measure, purity, entropy, precision, recall, and fitness value. Each measure's best, average, worst, and Std values are listed. Also, the convergences curves are illustrated in Fig. 6. The proposed method IAOA is compared to ten algorithms: AOA, GWO, HHO, EO, MPA, MVO, PSO, SCA, SSA, and WOA.

Based on the results of DAS1, as in Table 6, the IAOA, WOA, SCA, EO, and MPA provide the same results and are ranked first in terms of average, worst, and Std measures. Whereas the HHO is ranked second in these measures, it ranks first in the best metric. The worst algorithms are MVO and PSO, respectively. In the f-measure measure, the HHO has the first rank in both average and worst measures, followed by MVO and SSA. The IAOA, WOA, SCA, EO, and MPA show the same performance and results; therefore, they are ranked fourth. The HHO is also ranked first in the Purity measure, followed by SSA, PSO, MVO, AOA, WOA, and GWO, respectively; whereas, the IAOA, WOA, SCA EO, and MPA show the same results and are ranked eighth. However, the HHO is ranked first in best, average, and worst; it obtains the Purity measure's largest Std value. In the entropy measure, the IAOA, WOA, SCA, EO, and MPA are ranked first and show the same performance, whereas the PSO records the best entropy value. In the precision and recall measure, the PSO and HHO the first ranks, respectively, based on the best values. The IAOA, WOA, SCA, EO, and MPA show the same results, ranking fifth and third, respectively. The IAOA achieves the first rank in the best and worst values in the fitness measure, followed by MPA, EO, SCA, GWO, and WOA. Whereas the MVO is ranked last; however, it reaches the lowest

Table 7
The results of the comparative methods using dataset 2 (DAS2).

Measure	Metric	Comparative algorithms										
		AOA	GWO	HHO	EO	MPA	MVO	PSO	SCA	SSA	WOA	IAOA
Accuracy	Best	0.6250	0.6000	0.6000	0.5125	0.5125	0.6125	0.2000	0.5375	0.5125	0.5125	0.5375
	Average	0.5875	0.5875	0.5500	0.5125	0.5125	0.6000	0.1688	0.5250	0.5125	0.5063	0.5250
	Worst	0.5500	0.5750	0.5000	0.5125	0.5125	0.5875	0.1375	0.5125	0.5125	0.5000	0.5125
	Std	0.0530	0.0177	0.0707	0.0000	0.0000	0.0177	0.0442	0.0177	0.0000	0.0088	0.0177
	Rank	2	3	4	7	7	1	11	5	7	10	5
F-measure	Best	0.6962	0.6774	0.6001	0.6100	0.6100	0.6128	0.3045	0.6296	0.5128	0.6100	0.6296
	Average	0.6677	0.6679	0.5501	0.6100	0.6100	0.6019	0.2650	0.6198	0.5127	0.5550	0.6198
	Worst	0.6393	0.6585	0.5000	0.6100	0.6100	0.5911	0.2254	0.6100	0.5125	0.5000	0.6100
	Std	0.0403	0.0134	0.0708	0.0000	0.0000	0.0154	0.0559	0.0139	0.0002	0.0777	0.0139
	Rank	2	1	9	5	5	7	11	3	10	8	3
Purity	Best	2.1286	0.8604	0.6253	2.5127	2.5127	0.6383	0.8750	1.2595	0.5928	2.5127	2.5127
	Average	1.5117	0.8538	0.5754	1.8861	1.8861	0.6300	0.7965	1.0962	0.5655	1.6345	1.8890
	Worst	0.8947	0.8472	0.5256	1.2595	1.2595	0.6217	0.7179	0.9329	0.5381	0.7564	6.8593
	Std	0.8725	0.0093	0.0705	1.6140	0.6140	0.0117	0.1111	0.2309	0.0387	0.9698	0.6544
	Rank	5	7	10	2	2	9	8	6	11	4	1
Entropy	Best	0.2137	0.2435	0.3381	0.5197	0.5197	0.3901	0.3796	0.2690	0.3501	0.3712	0.2956
	Average	2.3797	0.2478	0.3663	0.7519	0.5519	0.3949	0.4023	0.3444	0.3949	0.4020	0.3022
	Worst	4.5458	0.2521	0.3945	0.8683	0.5987	0.3996	0.4377	0.5197	0.3982	0.4683	0.3255
	Std	0.0633	0.0061	0.0399	0.6053	0.6053	0.0067	0.2103	0.2480	0.0067	0.8518	0.0399
	Rank	11	1	4	10	9	5	8	3	5	7	2
Precision	Best	0.7857	0.7778	0.6003	0.7532	0.7532	0.6131	0.6378	0.7597	0.5132	0.7532	0.7597
	Average	0.7744	0.7740	0.5501	0.7532	0.7532	0.6039	0.6314	0.7565	0.5128	0.6266	0.7565
	Worst	0.7632	0.7703	0.5000	0.7532	0.7532	0.5947	0.6250	0.7532	0.5125	0.5000	0.7532
	Std	0.0159	0.0053	0.0709	0.0000	0.0000	0.0131	0.0091	0.0046	0.0005	0.1790	0.0046
	Rank	1	2	10	5	5	9	7	3	11	8	3
Recall	Best	0.6250	0.6000	0.6000	0.5125	0.5125	0.6125	0.2000	0.5375	0.5125	0.5125	0.5375
	Average	0.5875	0.5875	0.5500	0.5125	0.5125	0.6000	0.1688	0.5250	0.5125	0.5063	0.5250
	Worst	0.5500	0.5750	0.5000	0.5125	0.5125	0.5875	0.1375	0.5125	0.5125	0.5000	0.5125
	Std	0.0530	0.0177	0.0707	0.0000	0.0000	0.0177	0.0442	0.0177	0.0000	0.0088	0.0177
	Rank	2	3	4	7	7	1	11	5	7	10	5
Fitness	Best	0.3983	0.4139	0.2710	0.4560	0.4787	0.2763	0.3531	0.4777	0.2828	0.4022	0.4925
	Average	0.3895	0.4122	0.2703	0.4439	0.4281	0.2756	0.3408	0.4353	0.2827	0.3906	0.4866
	Worst	0.3806	0.4104	0.2696	0.4317	0.3775	0.2748	0.3286	0.3929	0.2826	0.3789	0.4807
	Std	0.0125	0.0025	0.0010	0.0172	0.0716	0.0011	0.0173	0.0600	0.0001	0.0165	0.0084
	Rank	7	5	11	2	4	10	8	3	9	6	1
Mean	Rank	4.29	3.14	7.43	5.43	5.57	6.00	9.14	4.00	8.57	7.57	2.86
Final	Ranking	4	2	8	5	6	7	11	3	10	9	1

Table 8
The results of the comparative methods using dataset 3 (DAS3).

Measure	Metric	Comparative algorithms										
		AOA	GWO	HHO	EO	MPA	MVO	PSO	SCA	SSA	WOA	IAOA
Accuracy	Best	0.3917	0.3500	0.4000	0.3500	0.3500	0.3667	0.1917	0.1917	0.3750	0.3583	0.3500
	Average	0.3750	0.3500	0.3958	0.3500	0.3458	0.3625	0.1750	0.1750	0.3708	0.3542	0.3458
	Worst	0.3583	0.3500	0.3917	0.3500	0.3417	0.3583	0.1583	0.1583	0.3667	0.3500	0.3417
	Std	0.0236	0.0000	0.0059	0.0000	0.0059	0.0059	0.0236	0.0236	0.0059	0.0059	0.0059
	Rank	2	6	1	6	8	4	10	10	3	5	8
F-measure	Best	0.3390	0.4831	0.4146	0.4831	0.4831	0.3632	0.2711	0.2711	0.3033	0.4831	0.4404
	Average	0.3209	0.4379	0.4030	0.4831	0.4351	0.3547	0.2515	0.2515	0.2991	0.4751	0.4319
	Worst	0.3029	0.3927	0.3915	0.4831	0.3870	0.3462	0.2319	0.2319	0.2949	0.4671	0.4234
	Std	0.0255	0.0639	0.0163	0.0000	0.0679	0.0120	0.0278	0.0278	0.0059	0.0114	0.0120
	Rank	8	3	6	1	4	7	10	10	9	2	5
Purity	Best	0.7773	1.3390	0.6954	1.3390	1.3390	0.4913	1.0563	1.0563	0.2665	2.8975	2.8923
	Average	0.7245	1.5904	0.5572	1.3390	1.0056	0.4536	0.9822	0.9822	0.2622	1.6733	1.6801
	Worst	0.0718	1.0842	0.4190	1.3390	0.9672	0.4159	0.9081	0.9081	0.2579	1.4492	1.5191
	Std	0.4989	5.3013	0.1954	0.0000	0.4714	0.0534	0.1048	0.1048	0.0061	0.8023	0.8326
	Rank	8	3	9	4	5	10	6	6	11	2	1
Entropy	Best	0.1647	0.5073	0.1521	0.4645	0.4855	0.3168	0.1346	0.1346	0.2705	0.7998	0.1325
	Average	0.4557	0.5373	0.2897	0.4957	0.5055	0.3501	0.1676	0.1676	0.2740	0.9063	0.1615
	Worst	0.7466	0.5674	0.4273	0.5053	0.5533	0.3835	0.2007	0.2007	0.2774	0.9958	0.2357
	Std	0.4115	0.3993	0.1946	0.7543	0.7543	0.0472	0.0467	0.0467	0.0048	0.5011	2.0284
	Rank	7	10	5	8	9	6	2	2	4	11	1
	Best	0.2988	0.7797	0.4302	0.7797	0.7797	0.3598	0.4632	0.4632	0.2546	0.7797	0.5936
	Average	0.2806	0.6135	0.4108	0.7797	0.6130	0.3473	0.4481	0.4481	0.2506	0.7251	0.5751

(continued on next page)

Table 8 (*continued*).

Measure	Metric	Comparative algorithms										
		AOA	GWO	HHO	EO	MPA	MVO	PSO	SCA	SSA	WOA	IAOA
Precision	Worst	0.2623	0.4473	0.3913	0.7797	0.4463	0.3348	0.4329	0.4329	0.2467	0.6705	0.5567
	Std	0.0258	0.2350	0.0275	0.0000	0.2357	0.0177	0.0214	0.0214	0.0056	0.0772	0.0261
	Rank	10	3	8	1	4	9	6	6	11	2	5
Recall	Best	0.3917	0.3500	0.4000	0.3500	0.3500	0.3667	0.1917	0.1917	0.3750	0.3583	0.3600
	Average	0.3750	0.3500	0.3958	0.3500	0.3458	0.3625	0.1750	0.1750	0.3708	0.3542	0.3558
	Worst	0.3583	0.3500	0.3917	0.3500	0.3417	0.3583	0.1583	0.1583	0.3667	0.3500	0.3402
	Std	0.0236	0.0000	0.0059	0.0000	0.0059	0.0059	0.0236	0.0236	0.0059	0.0059	0.0059
	Rank	2	7	1	7	9	4	10	10	3	6	5
Fitness	Best	0.3148	0.3400	0.2142	0.3997	0.3556	0.2088	0.2688	0.2688	0.2931	0.3292	0.4144
	Average	0.3130	0.3207	0.2091	0.3922	0.3534	0.2076	0.2648	0.2648	0.2755	0.3185	0.3975
	Worst	0.3111	0.3015	0.2040	0.3848	0.3511	0.2063	0.2607	0.2607	0.2579	0.3078	0.3805
	Std	0.0026	0.0273	0.0072	0.0106	0.0032	0.0017	0.0057	0.0057	0.0249	0.0151	0.0239
	Rank	6	4	10	2	3	11	8	8	7	5	1
Mean	Rank	6.14	5.14	5.71	4.14	6.00	7.29	7.43	7.43	6.86	4.71	3.71
Final	Ranking	7	4	5	2	6	9	10	10	8	3	1

Table 9
The results of the comparative methods using dataset 4 (DAS4).

Measure	Metric	Comparative algorithms										
		AOA	GWO	HHO	EO	MPA	MVO	PSO	SCA	SSA	WOA	IAOA
Accuracy	Best	0.3125	0.2813	0.3000	0.2750	0.2688	0.2938	0.1188	0.3313	0.3375	0.3000	0.3188
	Average	0.3000	0.2813	0.3000	0.2688	0.2688	0.2813	0.0969	0.3000	0.3094	0.2844	0.3031
	Worst	0.2875	0.2813	0.3000	0.2625	0.2688	0.2688	0.0750	0.2688	0.2813	0.2688	0.2875
	Std	0.0177	0.0000	0.0000	0.0088	0.0000	0.0177	0.0309	0.0442	0.0398	0.0221	0.0221
	Rank	3	7	3	9	10	7	11	3	1	6	2
F-measure	Best	0.3710	0.3474	0.3282	0.3582	0.4040	0.2948	0.1868	0.4040	0.3218	0.4040	0.3588
	Average	0.3531	0.3455	0.3135	0.3205	0.3841	0.2801	0.1520	0.3672	0.2727	0.3748	0.3487
	Worst	0.3351	0.3437	0.2988	0.2827	0.3641	0.2654	0.1172	0.3303	0.2236	0.3456	0.3385
	Std	0.0253	0.0027	0.0207	0.0534	0.0283	0.0207	0.0492	0.0522	0.0694	0.0414	0.0144
	Rank	4	6	8	7	1	9	11	3	10	2	5
Purity	Best	0.5956	2.3158	0.9855	1.7532	1.2532	0.4069	0.7782	1.2532	1.0621	2.2532	2.0963
	Average	0.5712	1.9693	0.6830	0.5366	1.1441	0.3834	0.7441	0.8775	1.0334	1.9256	1.8434
	Worst	0.5467	1.6227	0.3804	0.3200	0.9628	0.3599	0.7100	0.5019	1.0046	0.5980	0.5904
	Std	0.0345	6.1470	0.4279	0.3773	0.9775	0.0332	0.0482	7.6023	0.0407	7.5344	4.6004
	Rank	9	1	8	10	4	11	7	6	5	2	3
Entropy	Best	0.2174	4.0961	0.3926	0.3118	0.6261	0.3604	0.1432	0.4225	0.4805	0.2115	0.5422
	Average	0.2177	0.7229	0.7011	0.6927	0.8374	0.3740	0.2307	0.9182	1.0221	0.8127	0.5434
	Worst	0.2181	0.9480	1.0097	0.8651	0.9414	0.3876	0.3182	1.4140	1.5637	0.9414	0.5447
	Std	0.0004	0.9564	0.4364	0.9053	0.9407	0.0192	0.1238	1.3529	0.7660	0.9502	0.0742
	Rank	1	7	6	5	9	3	2	10	11	8	4
Precision	Best	0.4563	0.4543	0.3621	0.5637	0.8137	0.2958	0.4374	0.8137	0.3074	0.8137	0.4114
	Average	0.4290	0.4480	0.3299	0.4273	0.6889	0.2790	0.3526	0.5715	0.2465	0.6106	0.4110
	Worst	0.4017	0.4417	0.2977	0.2909	0.5641	0.2622	0.2677	0.3293	0.1856	0.4074	0.4105
	Std	0.0386	0.0090	0.0456	0.1929	0.1765	0.0238	0.1200	0.3425	0.0862	0.2873	0.0007
	Rank	5	4	9	6	1	10	8	3	11	2	7
Recall	Best	0.3125	0.2813	0.3000	0.2750	0.2688	0.2938	0.1188	0.3313	0.3375	0.3000	0.3188
	Average	0.3000	0.2813	0.3000	0.2688	0.2688	0.2813	0.0969	0.3000	0.3094	0.2844	0.3031
	Worst	0.2875	0.2813	0.3000	0.2625	0.2688	0.2688	0.0750	0.2688	0.2813	0.2688	0.2875
	Std	0.0177	0.0000	0.0000	0.0088	0.0000	0.0177	0.0309	0.0442	0.0398	0.0221	0.0221
	Rank	4	7	3	9	10	7	11	5	1	6	2
Fitness	Best	0.3139	0.3729	0.2189	0.4168	0.3575	0.2071	0.2688	0.3889	0.2842	0.3609	0.3286
	Average	0.3094	0.3614	0.2104	0.3754	0.3537	0.2050	0.2643	0.3113	0.2822	0.3156	0.3236
	Worst	0.3049	0.3499	0.2019	0.3339	0.3499	0.2029	0.2598	0.2338	0.2803	0.2704	0.3186
	Std	0.0064	0.0162	0.0120	0.0587	0.0054	0.0029	0.0063	0.1097	0.0028	0.0640	0.0071
	Rank	7	2	10	1	3	11	9	6	8	5	4
Mean	Rank	4.71	4.86	6.71	6.71	5.43	8.29	8.43	5.14	6.71	4.43	3.86
Final	Ranking	3	4	7	7	6	10	11	5	7	2	1

Std value. From the average values in all measures, the IAOA is ranked first, followed by MPA, EO, SCA, HHO, WOA, MVO, and SSA, respectively.

The results of the DAS2 are tabulated in Table 7; this table shows that the IAOA obtained the best values in both purity and fitness measures. The MVO attains the best values in both accuracy and recall. For the precision metric, the AOA locates in the first position. In terms of f-measure and purity, the GWO achieves the first rank in both measures. From the average values for all metrics, the IAOA has the first rank, followed by GWO, SCA, AOA, EO, MPA, and MVO, respectively. The WOA, SSA, and PSO show the worst performance. We concluded that the results show that the evaluation of the proposed IAOA is better in terms of the search balancing between exploration and exploitation.

In DAS3, as in Table 8, the IAOA gets the best values in 3 out of 7 measures, namely purity, entropy, and fitness. In contrast, the EO records the best f-measure and precision values. The HHO obtains the best values in two measures, i.e., accuracy and recall. The worst performers are attained by MVO, PSO, and SCA, respectively. Overall, the proposed IAOA is more stable than

Table 10
The results of the comparative methods using dataset 5 (DAS5).

Measure	Metric	Comparative algorithms										
		AOA	GWO	HHO	EO	MPA	MVO	PSO	SCA	SSA	WOA	IAOA
Accuracy	Best	0.2300	0.2350	0.2600	0.2150	0.2150	0.2900	0.0800	0.2800	0.2500	0.2350	0.2500
	Average	0.2225	0.2325	0.2525	0.2150	0.2125	0.2600	0.0750	0.2775	0.2375	0.2325	0.2475
	Worst	0.2150	0.2300	0.2450	0.2150	0.2100	0.2300	0.0700	0.2750	0.2250	0.2300	0.2450
	Std	0.0106	0.0035	0.0106	0.0000	0.0035	0.0424	0.0071	0.0035	0.0177	0.0035	0.0035
	Rank	8	6	3	9	10	2	11	1	5	6	4
F-measure	Best	0.3326	0.3135	0.2595	0.3220	0.3220	0.2892	0.1240	0.3342	0.2849	0.3188	0.3470
	Average	0.3161	0.3060	0.2523	0.3220	0.3032	0.2539	0.1174	0.3264	0.2815	0.3184	0.3087
	Worst	0.2996	0.2984	0.2452	0.3220	0.2845	0.2187	0.1108	0.3187	0.2781	0.3181	0.2705
	Std	0.0234	0.0106	0.0101	0.0000	0.0265	0.0499	0.0093	0.0110	0.0048	0.0005	0.0541
	Rank	4	6	10	2	7	9	11	1	8	3	5
Purity	Best	1.1093	2.6121	0.3631	0.9002	9.0020	0.3753	0.3782	0.8151	0.8082	5.4542	6.6313
	Average	1.0088	1.7374	0.3443	0.9002	4.9020	0.3651	0.3716	0.7383	0.6344	3.1743	6.4894
	Worst	0.9084	0.8626	0.3255	0.9002	0.8020	0.3548	0.3650	0.6614	0.4607	0.8945	6.3474
	Std	0.1420	1.2371	0.0266	0.0000	5.7983	0.0145	0.0093	0.1087	0.9074	3.2242	0.2007
	Rank	5	4	11	6	2	10	9	7	8	3	1
Entropy	Best	0.2839	0.4759	0.3826	0.9486	0.5482	0.3096	0.4241	0.0474	0.5366	0.4181	0.2810
	Average	0.3396	0.9883	0.3833	0.9486	0.7984	0.3747	0.4318	0.1707	0.9650	0.9392	0.3324
	Worst	0.3953	1.5007	0.3841	0.9486	0.9049	0.4398	0.4394	0.2941	1.1286	1.2366	0.3911
	Std	0.0788	1.8668	0.0010	0.0000	0.5486	0.0920	0.0108	0.1745	0.7394	0.8750	0.0788
	Rank	3	11	5	9	7	4	6	1	10	8	2
Precision	Best	0.6007	0.4921	0.2590	0.6408	0.6408	0.2885	0.2757	0.4259	0.3882	0.5193	0.5669
	Average	0.5473	0.4504	0.2522	0.6408	0.5408	0.2484	0.2709	0.3978	0.3508	0.5056	0.4344
	Worst	0.4940	0.4088	0.2453	0.6408	0.4408	0.2084	0.2661	0.3697	0.3134	0.4919	0.3019
	Std	0.0754	0.0589	0.0097	0.0000	0.1414	0.0566	0.0068	0.0397	0.0529	0.0194	0.1874
	Rank	2	5	10	1	3	11	9	7	8	4	6
Recall	Best	0.2300	0.2350	0.2600	0.2150	0.2150	0.2900	0.0800	0.2800	0.2500	0.2350	0.2500
	Average	0.2225	0.2325	0.2525	0.2150	0.2125	0.2600	0.0750	0.2775	0.2375	0.2325	0.2475
	Worst	0.2150	0.2300	0.2450	0.2150	0.2100	0.2300	0.0700	0.2750	0.2250	0.2300	0.2450
	Std	0.0106	0.0035	0.0106	0.0000	0.0035	0.0424	0.0071	0.0035	0.0177	0.0035	0.0035
	Rank	8	6	3	9	10	2	11	1	5	6	4
Fitness	Best	0.3698	0.3984	0.2234	0.5018	0.4572	0.2283	0.3327	0.2688	0.3674	0.4008	0.4025
	Average	0.3694	0.3904	0.2224	0.4823	0.4341	0.2271	0.3187	0.2656	0.3322	0.3733	0.3818
	Worst	0.3689	0.3824	0.2214	0.4628	0.4110	0.2258	0.3048	0.2624	0.2970	0.3457	0.3611
	Std	0.0006	0.0113	0.0014	0.0276	0.0327	0.0018	0.0198	0.0045	0.0497	0.0390	0.0293
	Rank	6	3	11	1	2	10	8	9	7	5	4
Mean	Rank	5.14	5.86	7.57	5.29	5.86	6.86	9.29	3.86	7.29	5.00	3.71
Final	Ranking	4	6	10	5	6	8	11	2	9	3	1

the other methods, and it always got better results than other methods.

For DAS4, Table 9 shows the obtained results by the implemented algorithms. From the recorded results, one can see that the SSA records the best values in both accuracy and recall while the MPA reaches the best results in both f-measure and precision. The AOA, GWO, and EO give the best results in entropy, purity, and fitness, respectively. However, the IAOA do not achieve the first rank in that measure, but it provided the best position in all measures' average. Thus, the effectiveness of using the OBL and LFD with AOA is excellent, and it can solve complicated problems as observed. The best solutions that have been obtained by the proposed method proved that the ability of the AOA is much better than the other well-known methods.

In DAS5, as in Table 10, the IAOA gets the best purity value and shows good performance in the rest measures. At the same time, the SCA is ranked first in the accuracy, f-measure, entropy, and recall measures. The EO reaches the best results in both precision and fitness for this dataset. Overall, the proposed IAOA got the first ranking over all the tested methods using the evaluation measures. So, this is clear evidence that the behavior of the proposed IAOA is more sable than the others, and it got the distribution results with a shorter range compared to all the other methods.

Furthermore, Table 11 records the results of the DAS6. As shown in Table 11, the IAOA gives the best results in three measures, including f-measure, precision, and fitness. At the same time, the HHO gets the best results in accuracy, recall, and entropy. SSA records the best purity value. The MPA shows the same performance as IAOA in f-measure and precision.

Also, the results of the DAS7 are listed in Table 12. In DAS7, the IAOA obtains the best f-measure value and is ranked first based on the purity measure. MVO gives the best results for accuracy and recall. The best precision, entropy, and fitness are recorded by EO, WOA, and MPA, respectively. The worst algorithm in all measures for this dataset is the PSO.

Besides, in DAS8, the PSO also records the worst performance in all measures, as shown in Table 13, while the MPA offers the best performance in purity, precision, and fitness measures. The IAOA records the best f-measure value, and the MVO registers the best accuracy and recall values. The HHO gets the entropy best value.

Moreover, Table 14 tabulates the results of the DAS9; the IAOA, in this dataset, reaches the best value in purity measure. At the same time, MVO reports the best values of the accuracy and recall measures. The MPA, EO, and GWO reach the first ranks in fitness, precision, and f-measure. Both SCA and SSA show the same performance in the entropy measure.

As in Table 15, the IAOA shows the first rank in the accuracy measure in DAS10, and the MPA achieves the first rank in entropy, precision, and fitness measure. The AOA, EO, and MVO are ranked first in f-measure, purity, and recall, respectively. The worst performers are shown by both WOA and PSO. Overall, the proposed IAOA got the best ranking compared with other well-known tested methods the evaluation measures. It is clear here

Table 11
The results of the comparative methods using dataset 6 (DAS6).

Measure	Metric	Comparative algorithms										
		AOA	GWO	HHO	EO	MPA	MVO	PSO	SCA	SSA	WOA	IAOA
Accuracy	Best	0.2125	0.2083	0.2542	0.1917	0.1833	0.2000	0.0625	0.2208	0.2250	0.2083	0.2255
	Average	0.2083	0.2063	0.2500	0.1917	0.1833	0.1958	0.0604	0.2188	0.2229	0.1979	0.2152
	Worst	0.2042	0.2042	0.2458	0.1917	0.1833	0.1917	0.0583	0.2167	0.2208	0.1875	0.2102
	Std	0.0059	0.0029	0.0059	0.0000	0.0000	0.0059	0.0029	0.0029	0.0029	0.0147	0.0124
	Rank	5	6	1	9	10	8	11	3	2	7	4
F-measure	Best	0.2889	0.2824	0.2595	0.2906	0.2901	0.2876	0.1015	0.3076	0.3031	0.2737	0.2901
	Average	0.2825	0.2583	0.2478	0.2763	0.2861	0.2506	0.0983	0.2685	0.2613	0.2384	0.2861
	Worst	0.2760	0.2342	0.2361	0.2620	0.2821	0.2135	0.0950	0.2294	0.2195	0.2031	0.2821
	Std	0.0091	0.0341	0.0165	0.0202	0.0057	0.0523	0.0046	0.0553	0.0591	0.0499	0.0057
	Rank	3	7	9	4	1	8	11	5	6	10	1
Purity	Best	2.1787	1.6988	0.4516	1.8824	1.3348	1.0619	0.4571	1.2314	2.0497	1.5467	1.3348
	Average	1.3321	1.2765	0.4353	1.2464	1.2514	0.6690	0.3700	0.9378	1.6186	0.9798	1.2514
	Worst	1.0854	0.8541	0.4191	0.6105	1.1681	0.2761	0.2828	0.6442	1.1876	0.4129	1.1681
	Std	2.6116	2.0115	0.0229	5.1420	0.1178	0.5557	0.1232	0.4152	0.6096	0.8018	0.1178
	Rank	2	3	10	6	4	9	11	8	1	7	4
Entropy	Best	0.5171	0.3608	0.3287	0.0266	0.5731	0.3814	0.4180	1.2352	0.9993	0.2233	0.5731
	Average	0.9139	0.6130	0.3361	0.8695	0.7464	0.5455	0.4445	1.4825	1.5426	0.5469	0.7464
	Worst	1.1057	0.8651	0.3435	0.7125	0.9198	0.7095	0.4710	1.7449	1.7086	0.8706	0.9198
	Std	0.9952	0.9840	0.0105	0.9155	0.2451	0.2320	0.0375	0.8056	0.9425	0.8616	0.2451
	Rank	9	5	1	8	6	3	2	10	11	4	6
Precision	Best	0.4939	0.4382	0.2651	0.6005	0.6950	0.5116	0.2699	0.5069	0.4830	0.5067	0.6950
	Average	0.4438	0.3564	0.2461	0.5072	0.6534	0.3763	0.2631	0.3753	0.3486	0.3524	0.6534
	Worst	0.3937	0.2747	0.2272	0.4139	0.6118	0.2411	0.2563	0.2438	0.2142	0.1982	0.6118
	Std	0.0709	0.1157	0.0268	0.1320	0.0588	0.1913	0.0096	0.1860	0.1901	0.2182	0.0588
	Rank	4	7	11	3	1	5	10	6	9	8	1
Recall	Best	0.2125	0.2083	0.2542	0.1917	0.1844	0.2000	0.0625	0.2208	0.2250	0.2083	0.1833
	Average	0.2083	0.2063	0.2500	0.1917	0.1844	0.1958	0.0604	0.2188	0.2229	0.1979	0.1833
	Worst	0.2042	0.2042	0.2458	0.1917	0.1844	0.1917	0.0583	0.2167	0.2208	0.1875	0.1833
	Std	0.0059	0.0029	0.0059	0.0000	0.0000	0.0059	0.0029	0.0029	0.0029	0.0147	0.0000
	Rank	4	5	1	8	9	7	11	3	2	6	10
Fitness	Best	0.3888	0.3915	0.2232	0.4167	0.4125	0.3522	0.3387	0.3683	0.3108	0.3497	0.4266
	Average	0.3747	0.3768	0.2217	0.4151	0.4157	0.3455	0.3091	0.3089	0.3059	0.3345	0.4253
	Worst	0.3606	0.3621	0.2201	0.4135	0.4240	0.3389	0.2795	0.2496	0.3010	0.3193	0.4240
	Std	0.0199	0.0208	0.0022	0.0022	0.0018	0.0095	0.0419	0.0839	0.0069	0.0215	0.0018
	Rank	5	4	11	3	2	6	8	9	10	7	1
Mean	Rank	4.57	5.29	6.29	5.86	4.71	6.57	9.14	6.29	5.86	7.00	3.86
Final	Ranking	2	4	7	5	3	9	11	7	5	10	1

Table 12
The results of the comparative methods using dataset 7 (DAS7).

Measure	Metric	Comparative algorithms										
		AOA	GWO	HHO	EO	MPA	MVO	PSO	SCA	SSA	WOA	IAOA
Accuracy	Best	0.1893	0.1929	0.2107	0.1714	0.1571	0.2429	0.0536	0.2321	0.1857	0.1714	0.2000
	Average	0.1893	0.1804	0.2071	0.1679	0.1536	0.2286	0.0464	0.2196	0.1839	0.1661	0.1982
	Worst	0.1893	0.1679	0.2036	0.1643	0.1500	0.2143	0.0393	0.2071	0.1821	0.1607	0.1964
	Std	0.0000	0.0177	0.0051	0.0051	0.0051	0.0202	0.0101	0.0177	0.0025	0.0076	0.0025
	Rank	5	7	3	8	10	1	11	2	6	9	4
F-measure	Best	0.2904	0.2170	0.2063	0.2577	0.2484	0.2469	0.0896	0.2475	0.2195	0.2332	0.2696
	Average	0.2553	0.2072	0.2052	0.2545	0.2233	0.2291	0.0807	0.2350	0.2129	0.2132	0.2672
	Worst	0.2202	0.1973	0.2041	0.2514	0.1983	0.2112	0.0717	0.2225	0.2064	0.1931	0.2648
	Std	0.0496	0.0140	0.0016	0.0045	0.0354	0.0252	0.0126	0.0176	0.0093	0.0284	0.0034
	Rank	2	9	10	3	6	5	11	4	8	7	1
Purity	Best	1.8623	2.0793	0.8467	1.5149	1.1439	0.2941	0.7755	1.0479	1.5812	0.6677	2.0325
	Average	1.4896	1.3680	0.5561	1.2282	0.8582	0.2858	0.5776	0.7959	1.2142	0.4602	1.7580
	Worst	1.1170	0.6567	0.2656	1.1096	0.5725	0.2774	0.3798	0.5439	0.8471	0.2527	0.4835
	Std	2.3555	1.0059	0.4109	3.8080	0.4041	0.0118	0.2798	0.3564	0.5191	0.2935	2.2166
	Rank	2	3	9	4	6	11	8	7	5	10	1
Entropy	Best	1.6432	0.6561	0.4151	0.2573	0.3928	0.4048	0.2798	0.3614	0.9627	0.0099	0.3416
	Average	1.2620	0.8523	0.9192	1.3293	0.5908	0.4528	0.4385	0.4246	1.1636	0.2080	1.3110
	Worst	1.8808	1.0485	1.4233	1.5401	0.7889	0.5007	0.5971	0.4879	1.3645	0.4060	1.8784
	Std	1.8579	5.9343	0.7129	1.8541	0.2801	0.0678	0.2244	0.0895	1.5267	0.2801	1.9689
	Rank	9	6	7	11	5	4	3	2	8	1	10
Precision	Best	0.6215	0.2475	0.2041	0.5732	0.5923	0.2510	0.4026	0.2677	0.2761	0.4256	0.4112
	Average	0.4424	0.2441	0.2035	0.5625	0.4494	0.2295	0.3370	0.2544	0.2541	0.3219	0.4103
	Worst	0.2633	0.2406	0.2028	0.5517	0.3066	0.2080	0.2714	0.2410	0.2321	0.2181	0.4094
	Std	0.2533	0.0049	0.0009	0.0152	0.2020	0.0304	0.0928	0.0189	0.0311	0.1467	0.0013

(continued on next page)

Table 12 (*continued*).

Measure	Metric	Comparative algorithms										
		AOA	GWO	HHO	EO	MPA	MVO	PSO	SCA	SSA	WOA	IAOA
	Rank	3	9	11	1	2	10	5	7	8	6	4
Recall	Best	0.1895	0.1932	0.2099	0.1681	0.1571	0.2430	0.0537	0.2301	0.1858	0.1732	0.2005
	Average	0.1893	0.1802	0.2069	0.1645	0.1518	0.2288	0.0465	0.2184	0.1840	0.1669	0.1981
	Worst	0.1892	0.1672	0.2040	0.1610	0.1465	0.2145	0.0394	0.2066	0.1822	0.1606	0.1957
	Std	0.0002	0.0184	0.0041	0.0050	0.0075	0.0202	0.0101	0.0166	0.0026	0.0089	0.0035
	Rank	5	7	3	9	10	1	11	2	6	8	4
Fitness	Best	0.4261	0.3512	0.2657	0.4091	0.4603	0.2200	0.3139	0.2511	0.2874	0.3818	0.3841
	Average	0.3776	0.3469	0.2402	0.4049	0.4373	0.2185	0.3119	0.2509	0.2874	0.3505	0.3766
	Worst	0.3290	0.3426	0.2147	0.4007	0.4143	0.2170	0.3098	0.2507	0.2873	0.3192	0.3691
	Std	0.0686	0.0061	0.0361	0.0060	0.0325	0.0022	0.0030	0.0003	0.0001	0.0443	0.0107
	Rank	3	6	10	2	1	11	7	9	8	5	4
Mean	Rank	4.14	6.71	7.57	5.43	5.71	6.14	8.00	4.71	7.00	6.57	4.00
Final	Ranking	2	8	10	4	5	6	11	3	9	7	1

Table 13
The results of the comparative methods using dataset 8 (DAS8).

Measure	Metric	Comparative algorithms										
		AOA	GWO	HHO	EO	MPA	MVO	PSO	SCA	SSA	WOA	IAOA
Accuracy	Best	0.1594	0.1875	0.1813	0.1719	0.1469	0.2156	0.0500	0.1906	0.1813	0.1500	0.1781
	Average	0.1563	0.1750	0.1766	0.1656	0.1438	0.2078	0.0453	0.1891	0.1750	0.1453	0.1688
	Worst	0.1531	0.1625	0.1719	0.1594	0.1406	0.2000	0.0406	0.1875	0.1688	0.1406	0.1594
	Std	0.0044	0.0177	0.0066	0.0088	0.0044	0.0110	0.0066	0.0022	0.0088	0.0066	0.0133
	Rank	8	4	3	7	10	1	11	2	4	9	6
F-measure	Best	0.2408	0.2273	0.1707	0.2507	0.2368	0.2165	0.0854	0.2171	0.2243	0.2425	0.2547
	Average	0.2378	0.2260	0.1642	0.2447	0.2337	0.2086	0.0781	0.2069	0.2170	0.2220	0.2450
	Worst	0.2348	0.2246	0.1577	0.2387	0.2305	0.2007	0.0707	0.1968	0.2097	0.2016	0.2354
	Std	0.0043	0.0019	0.0092	0.0085	0.0044	0.0112	0.0104	0.0143	0.0103	0.0289	0.0136
	Rank	3	5	10	2	4	8	11	9	7	6	1
Purity	Best	2.0143	1.5566	0.1854	1.9152	2.1571	0.2806	0.6787	0.6345	1.8617	1.8653	1.8018
	Average	1.3916	1.2325	0.1702	1.4668	1.9704	0.2716	0.5373	0.4325	1.2549	1.5916	1.4901
	Worst	1.7689	1.1095	0.1551	1.3206	1.2508	0.2626	0.3958	0.2304	0.6481	0.8967	1.1784
	Std	0.9482	2.9532	0.0214	0.4912	3.4693	0.0127	0.2000	0.2857	0.8581	3.8112	1.8550
	Rank	5	7	11	4	1	10	8	9	6	2	3
Entropy	Best	0.9508	0.5165	0.3049	1.3278	0.8639	0.4412	0.4370	0.2535	0.8211	0.7135	0.7651
	Average	1.4515	1.2117	0.3101	1.9564	1.5343	0.4598	0.5599	0.3363	1.8401	0.9725	0.9240
	Worst	1.5226	1.9068	0.3154	2.0635	1.6205	0.4784	0.6828	0.4191	2.0469	1.2436	1.4774
	Std	0.9206	1.1080	0.0074	1.2273	1.3572	0.0263	0.1738	0.1171	1.4830	1.5156	1.6907
	Rank	8	7	1	11	9	3	4	2	10	6	5
Precision	Best	0.5262	0.3511	0.1614	0.4736	0.6410	0.2177	0.3014	0.2549	0.3349	0.6324	0.4503
	Average	0.5207	0.3196	0.1536	0.4689	0.6256	0.2094	0.2818	0.2320	0.2923	0.4944	0.4491
	Worst	0.5153	0.2881	0.1459	0.4643	0.6103	0.2011	0.2622	0.2092	0.2496	0.3564	0.4480
	Std	0.0077	0.0445	0.0109	0.0066	0.0217	0.0118	0.0277	0.0323	0.0603	0.1952	0.0016
	Rank	2	6	11	4	1	10	8	9	7	3	5
Recall	Best	0.1571	0.1877	0.1813	0.1717	0.1469	0.2153	0.0498	0.1890	0.1808	0.1500	0.1779
	Average	0.1541	0.1764	0.1765	0.1656	0.1437	0.2078	0.0453	0.1874	0.1747	0.1453	0.1686
	Worst	0.1511	0.1651	0.1717	0.1595	0.1405	0.2003	0.0409	0.1858	0.1686	0.1405	0.1594
	Std	0.0042	0.0160	0.0068	0.0086	0.0045	0.0106	0.0063	0.0023	0.0086	0.0067	0.0131
	Rank	8	4	3	7	10	1	11	2	5	9	6
Fitness	Best	0.3244	0.3718	0.2169	0.3833	0.3995	0.2150	0.3153	0.2557	0.3028	0.3727	0.3277
	Average	0.3182	0.3685	0.2165	0.3661	0.3863	0.2147	0.3081	0.2521	0.2925	0.3653	0.3253
	Worst	0.3119	0.3653	0.2162	0.3489	0.3731	0.2144	0.3010	0.2485	0.2822	0.3578	0.3228
	Std	0.0088	0.0046	0.0005	0.0244	0.0187	0.0004	0.0102	0.0051	0.0146	0.0105	0.0035
	Rank	6	2	10	3	1	11	7	9	8	4	5
Mean	Rank	5.71	5.00	7.00	5.43	5.14	6.29	8.57	6.00	6.71	5.57	4.43
Final	Ranking	6	2	10	4	3	8	11	7	9	5	1

that the ability of the IAOA in finding a promising search area better than the other, and it can obtain new best solutions.

From the results mentioned above, there are ten datasets with seven measures for each one (i.e., a total of 70 measures); these measures provide that the proposed method IAOA shows good performance and achieves the first rank in 17 out of 70 measures, followed by MPA and EO in 13 and 10 measures, respectively. Both HHO and MVO get the first rank in 9 measures, followed by SCA, SSA, GWO, and AOA. The worst algorithms in all measures are the WOA and PSO. Fig. 5 summarizes these results.

A nonparametric analysis, namely the Friedman test, is performed to evaluate the IAOA and the comparative algorithms' results. The test shows the significant differences between the algorithms via computing the p-value; if the p-value is less than 0.05, that is evidence of an existing difference between the compared algorithms. The Friedman test's statistical results for all algorithms are registered in Table 16 for ten datasets. The average ranks overall the datasets are given at the bottom of the table. From this table, we can conclude that the proposed method IAOA has the lowest rank in all datasets through the Fridman test; therefore, it is in the first position in the implemented algorithms' queue and followed by the EO, MPA, and GWO, respectively. The AOA and SCA have the fifth rank, followed by the WOA, SSA, and

Table 14
The results of the comparative methods using dataset 9 (DAS9).

Measure	Metric	Comparative algorithms										
		AOA	GWO	HHO	EO	MPA	MVO	PSO	SCA	SSA	WOA	IAOA
Accuracy	Best	0.1556	0.1833	0.1722	0.1583	0.1361	0.2028	0.0472	0.1667	0.1667	0.1500	0.1881
	Average	0.1542	0.1736	0.1694	0.1486	0.1347	0.1903	0.0431	0.1625	0.1625	0.1444	0.1856
	Worst	0.1528	0.1639	0.1667	0.1389	0.1333	0.1778	0.0389	0.1583	0.1583	0.1389	0.1745
	Std	0.0020	0.0137	0.0039	0.0137	0.0020	0.0177	0.0059	0.0059	0.0059	0.0079	0.0079
	Rank	7	3	4	8	10	1	11	5	5	9	2
F-measure	Best	0.1974	0.2641	0.1856	0.2393	0.2193	0.2067	0.0816	0.2163	0.2163	0.2084	0.2264
	Average	0.1922	0.2415	0.1745	0.2297	0.2058	0.1944	0.0760	0.2106	0.2106	0.2078	0.2195
	Worst	0.1871	0.2189	0.1635	0.2201	0.1922	0.1822	0.0704	0.2049	0.2049	0.2073	0.2127
	Std	0.0073	0.0319	0.0156	0.0135	0.0191	0.0173	0.0079	0.0080	0.0080	0.0008	0.0097
	Rank	9	1	10	2	7	8	11	4	4	6	3
Purity	Best	1.2697	1.3964	0.3243	1.3152	1.5636	0.2513	0.8642	0.7156	0.7156	1.8679	2.2993
	Average	1.0162	1.0272	0.2746	1.1960	1.3674	0.2404	0.6366	0.6303	0.6303	1.6777	1.9054
	Worst	0.7628	0.9006	0.2250	1.0769	5.1713	0.2295	0.4090	0.5450	0.5450	1.4874	1.5116
	Std	0.3584	2.3953	0.0702	0.1685	0.2774	0.0154	0.3218	0.1207	0.1207	0.2690	0.5570
	Rank	6	5	10	4	3	11	7	8	8	2	1
Entropy	Best	0.5305	0.8100	0.3733	0.7811	0.7153	0.4593	0.6557	0.0146	0.0146	3.9030	0.4253
	Average	1.1382	0.9562	0.4235	0.9853	0.9741	0.4795	0.7514	0.4233	0.4233	6.9183	0.5463
	Worst	1.7460	0.9903	0.4737	1.0835	1.4101	0.4997	0.8471	0.8321	0.8321	9.9336	0.9653
	Std	1.6880	1.4072	0.0710	1.2250	2.3607	0.0286	0.1354	0.5781	0.5781	4.2643	0.9564
	Rank	10	7	3	9	8	4	6	1	1	11	5
Precision	Best	0.2691	0.4734	0.2019	0.5291	0.6128	0.2103	0.3981	0.3131	0.3131	0.4437	0.3700
	Average	0.2540	0.4050	0.1811	0.5194	0.4886	0.1984	0.3463	0.3036	0.3036	0.3909	0.3676
	Worst	0.2389	0.3367	0.1603	0.5097	0.3643	0.1866	0.2944	0.2941	0.2941	0.3380	0.3651
	Std	0.0213	0.0967	0.0294	0.0137	0.1757	0.0167	0.0734	0.0134	0.0134	0.0748	0.0035
	Rank	9	3	11	1	2	10	6	7	7	4	5
Recall	Best	0.1559	0.1831	0.1717	0.1563	0.1336	0.2033	0.0474	0.1652	0.1652	0.1495	0.1631
	Average	0.1548	0.1726	0.1692	0.1477	0.1321	0.1906	0.0430	0.1612	0.1612	0.1428	0.1566
	Worst	0.1537	0.1622	0.1668	0.1390	0.1306	0.1780	0.0386	0.1572	0.1572	0.1362	0.1501
	Std	0.0015	0.0148	0.0034	0.0123	0.0021	0.0179	0.0062	0.0057	0.0057	0.0094	0.0092
	Rank	7	2	3	8	10	1	11	4	4	9	6
Fitness	Best	0.2673	0.3430	0.2082	0.3672	0.4044	0.2111	0.3347	0.2858	0.2858	0.3281	0.3824
	Average	0.2592	0.3413	0.2060	0.3557	0.3929	0.2088	0.3181	0.2804	0.2804	0.3277	0.3421
	Worst	0.2512	0.3396	0.2038	0.3443	0.3815	0.2066	0.3015	0.2749	0.2749	0.3273	0.3018
	Std	0.0114	0.0024	0.0031	0.0162	0.0162	0.0032	0.0235	0.0077	0.0077	0.0005	0.0570
	Rank	9	4	11	2	1	10	6	7	7	5	3
Mean	Rank	8.14	3.57	7.43	4.86	5.86	6.43	8.29	5.14	5.14	6.57	3.57
Final	Ranking	10	1	9	3	6	7	11	4	4	8	1

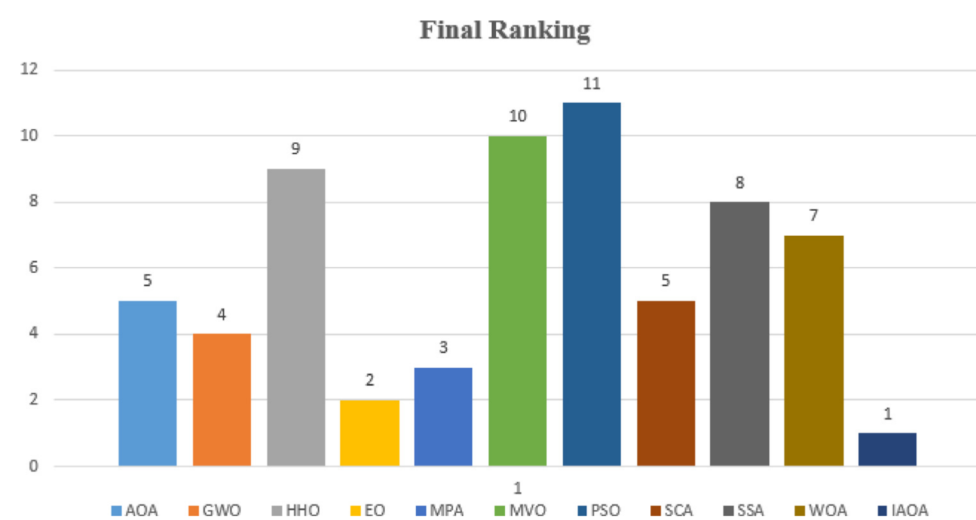

Fig. 5. The ranks for all the tested algorithms overall datasets.

HHO. In contrast, the MVO and PSO show the worst performance and locate the last position.

The last aspect that can be considered while evaluating the proposed IAOA is the convergence property of the algorithm; thus, the attained performance of the IAOA across the number of iterations versus the comparable algorithms are depicted in Fig. 6 for the studied ten datasets. The figure's convergence curves show that the IAOA can smoothly converge to high-quality solutions with fast behavior, confirming its efficiency and applicability for optimizing the document clustering optimization problem. The tested methods' convergence curves show that the proposed IAOA converges stably, and it never stuck in the local optima area. This proved that the incorporated methods (i.e., OBL and LFD) into the conventional AOA enhanced its ability in the

Table 15
The results of the comparative methods using dataset 10 (DAS10).

Measure	Metric	Comparative algorithms										
		AOA	GWO	HHO	EO	MPA	MVO	PSO	SCA	SSA	WOA	IAOA
Accuracy	Best	0.1450	0.1500	0.1700	0.1350	0.1275	0.1825	0.0400	0.1550	0.1525	0.1350	0.1925
	Average	0.1438	0.1438	0.1600	0.1350	0.1238	0.1750	0.0375	0.1538	0.1488	0.1338	0.1813
	Worst	0.1425	0.1375	0.1500	0.1350	0.1200	0.1675	0.0350	0.1525	0.1450	0.1325	0.1600
	Std	0.0018	0.0088	0.0141	0.0000	0.0053	0.0106	0.0035	0.0018	0.0053	0.0018	0.0018
	Rank	6	6	3	8	10	2	11	4	5	9	1
F-measure	Best	0.2258	0.2072	0.1741	0.2150	0.2019	0.1846	0.0732	0.1739	0.1864	0.2002	0.2106
	Average	0.2116	0.1831	0.1722	0.2078	0.1947	0.1756	0.0684	0.1720	0.1835	0.1971	0.2035
	Worst	0.1974	0.1591	0.1703	0.2007	0.1875	0.1666	0.0636	0.1701	0.1806	0.1940	0.1963
	Std	0.0201	0.0341	0.0027	0.0102	0.0102	0.0128	0.0068	0.0027	0.0041	0.0044	0.0101
	Rank	1	7	9	2	5	8	11	10	6	4	3
Purity	Best	2.3736	1.4171	0.3635	4.4836	2.4430	0.2294	0.7183	0.4204	0.8785	1.2622	2.4775
	Average	1.9871	1.1314	0.3126	2.9136	1.6892	0.2284	0.6292	0.3990	0.7265	0.8983	1.7232
	Worst	1.6006	0.8456	0.2618	1.3436	1.1354	0.2274	0.5400	0.3775	0.5746	0.5344	0.9688
	Std	0.5466	0.4041	0.0719	2.2204	0.3590	0.0014	0.1261	0.0304	0.2149	0.5146	1.0668
	Rank	2	5	10	1	4	11	8	9	7	6	3
Entropy	Best	4.6074	1.2392	0.3328	0.9424	1.7019	0.4141	1.1308	0.3263	0.1538	0.6076	0.3265
	Average	1.7440	1.4691	0.3845	1.2446	0.1943	0.4492	1.2400	0.3602	0.5814	1.0769	0.3522
	Worst	1.8806	1.9989	0.4362	1.3968	1.0620	0.4844	1.3493	0.3940	1.0090	1.5463	0.3851
	Std	0.5780	3.5778	0.0731	1.1271	1.7815	0.0497	0.1545	0.0479	0.6047	0.6637	0.5653
	Rank	11	10	4	9	1	5	8	3	6	7	2
Precision	Best	0.4634	0.3252	0.2081	0.4928	0.4827	0.1870	0.4183	0.1998	0.2388	0.4042	0.3900
	Average	0.3746	0.2562	0.1896	0.4396	0.4724	0.1762	0.3817	0.1975	0.2376	0.3916	0.3502
	Worst	0.2858	0.1873	0.1710	0.3864	0.4621	0.1655	0.3450	0.1953	0.2364	0.3790	0.3104
	Std	0.1256	0.0975	0.0263	0.0753	0.0145	0.0152	0.0519	0.0032	0.0017	0.0178	0.0563
	Rank	5	7	10	2	1	11	4	9	8	3	6
Recall	Best	0.1508	0.1521	0.1696	0.1375	0.1276	0.1823	0.0401	0.1539	0.1528	0.1331	0.1443
	Average	0.1500	0.1451	0.1596	0.1365	0.1226	0.1750	0.0376	0.1523	0.1495	0.1317	0.1439
	Worst	0.1492	0.1382	0.1496	0.1355	0.1176	0.1677	0.0350	0.1508	0.1461	0.1304	0.1436
	Std	0.0011	0.0098	0.0142	0.0014	0.0071	0.0103	0.0036	0.0023	0.0048	0.0019	0.0005
	Rank	4	6	2	8	10	1	11	3	5	9	7
Fitness	Best	0.3217	0.3433	0.2143	0.3587	0.3615	0.2057	0.3025	0.2413	0.2602	0.3058	0.3678
	Average	0.3156	0.3361	0.2081	0.3535	0.3593	0.2049	0.2878	0.2410	0.2600	0.2997	0.3563
	Worst	0.3095	0.3289	0.2019	0.3483	0.3572	0.2041	0.2732	0.2406	0.2598	0.2935	0.3171
	Std	0.0086	0.0102	0.0088	0.0073	0.0030	0.0011	0.0207	0.0005	0.0003	0.0087	0.0076
	Rank	5	4	10	3	1	11	7	9	8	6	2
Mean	Rank	4.86	6.43	6.86	4.71	4.57	7.00	8.57	6.71	6.43	6.29	3.43
Final	Ranking	4	6	9	3	2	10	11	8	6	5	1

Table 16
The final statistical results of the comparative methods using Friedman ranking test.

Dataset	Comparative algorithms										
	AOA	GWO	HHO	EO	MPA	MVO	PSO	SCA	SSA	WOA	IAOA
DAS1	11	10	5	3	2	7	9	4	7	6	1
DAS2	4	2	8	5	6	7	11	3	10	9	1
DAS3	7	4	5	2	6	9	10	10	8	3	1
DAS4	3	4	7	7	6	10	11	5	7	2	1
DAS5	4	6	10	5	6	8	11	2	9	3	1
DAS6	2	4	7	5	3	9	11	7	5	10	1
DAS7	2	8	10	4	5	6	11	3	9	7	1
DAS8	6	2	10	4	3	8	11	7	9	5	1
DAS9	10	1	9	3	6	7	11	4	4	8	1
DAS10	4	6	9	3	2	10	11	8	6	5	1
Mean rank	5.3	4.7	8.0	4.1	4.5	8.1	10.7	5.3	7.4	5.8	1.0
Final Ranking	5	4	9	2	3	10	11	5	8	7	1

search, keeping the solutions' diversity, finding new search areas, and properly balancing the search process.

4.2.4. Comparisons with the state-of-the-art methods

In this section, we tested the proposed IAOA using 31 different text datasets, and the obtained results are compared with 21 state-of-the-art methods published in the literature.

Table 17 shows the state-of-the-art comparative methods, and Table 18 shows the benchmark datasets used in this comparison to prove the effectiveness of the proposed IAOA compared to other previous methods. This table presents details of each dataset, including the number of documents, number of clusters, and their references. Table 19 illustrates that the proposed IAOA method's results compared with most state-of-the-art works using their datasets. This table includes the results of 21 methods tested on 31 datasets.

Table 19 presents the results of the proposed IAOA compared with several well-known clustering methods based on five standard evaluation measures typically utilized in the domain of text mining, including Precision, Recall, F-measure, Entropy, and Purity. This table divides the comparisons into 31 parts according to the selected datasets. The clustering results obtained by the evaluation criteria are given in each row for each dataset. The indicator "-" denotes the method that did not report a result

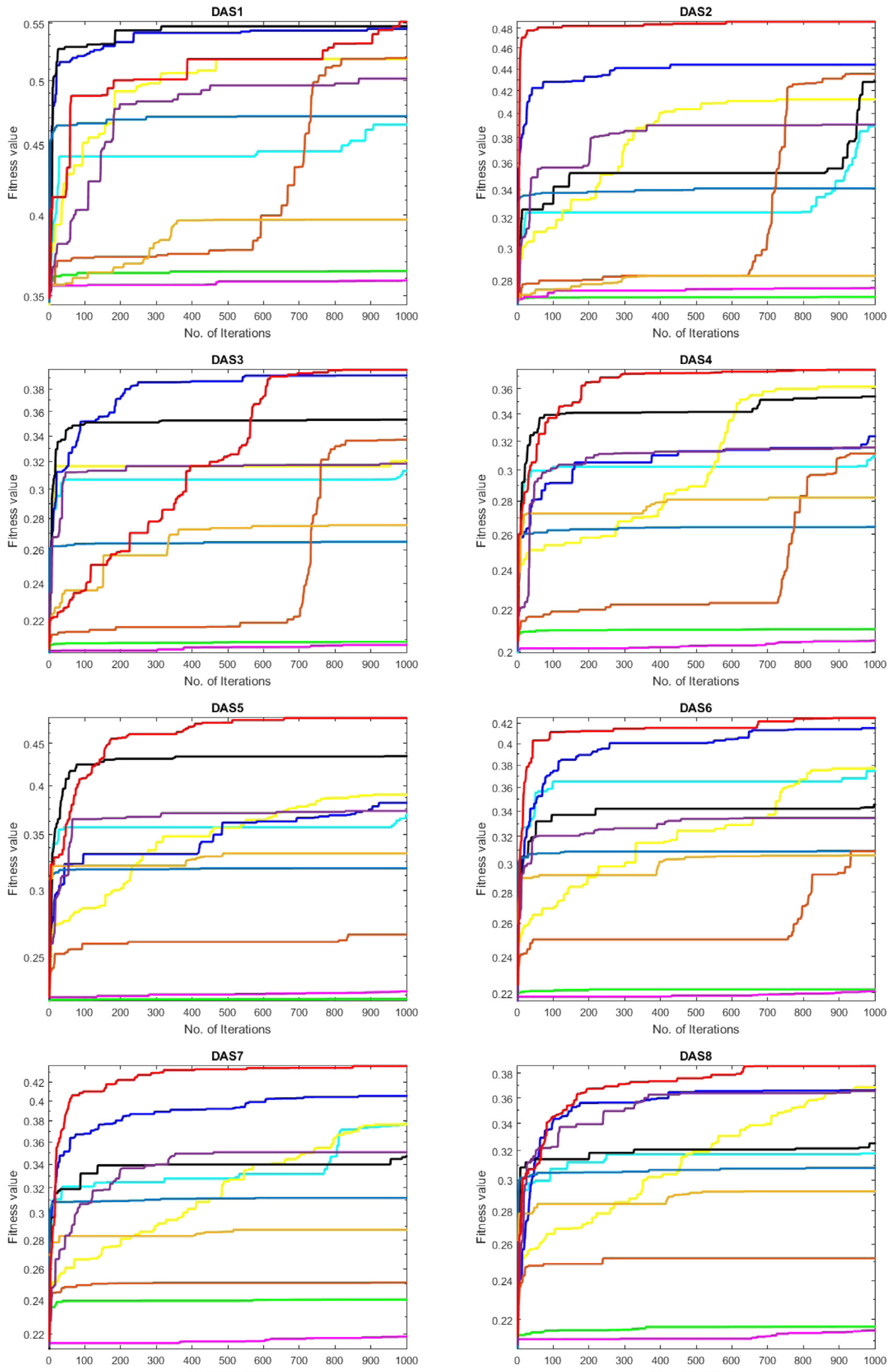

Fig. 6. Convergence behavior of the comparative methods on the tested datasets (DAS1–DAS10).

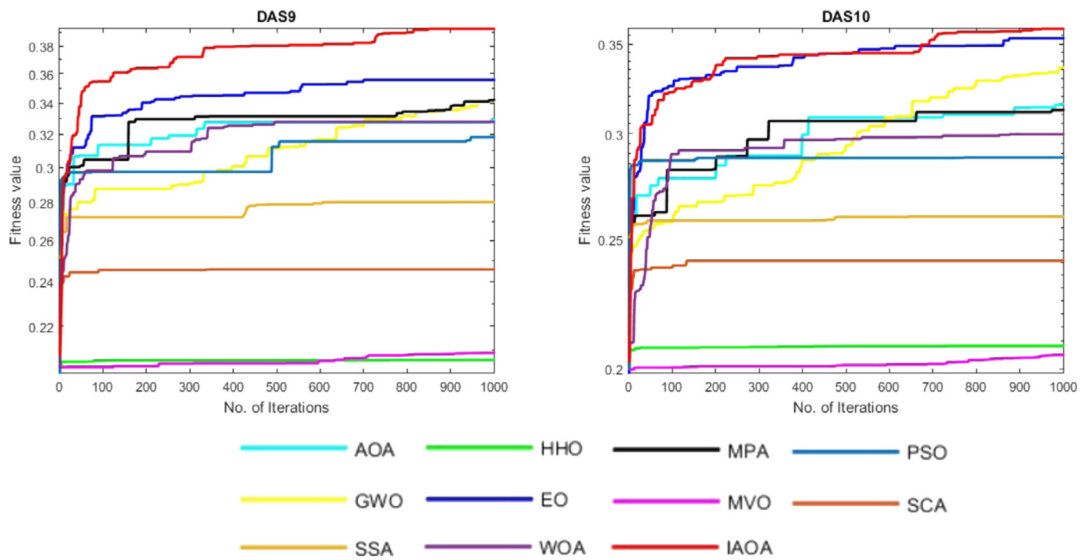

Fig. 6. (*continued*).

Table 17
Key of the comparative state-of-the-art methods.

No.	Method	Key	Year	Reference
1	Heuristic K-means	HK-means	2011	[90]
2	Hybrid Harmony Search with K-mean Algorithm	HS+k-mean	2013	[62]
3	Particle Swarm Optimization	KPSO	2013	[91]
4	Cuckoo Search Algorithm	CS	2013	[92]
5	Bisecting K-mean Algorithm	BK-mean	2014	[93]
6	An Improved Bee Colony Optimization Algorithm	IBCOCLUST	2015	[94]
7	Firefly Algorithm	WFA2	2015	[12]
8	K-mean Text Document Clustering	TV-DF+K-mean	2015	[50]
9	Chaotic Gradient Artificial Bee Colony	ABC	2016	[11]
10	K-mean Text Document Clustering	BPSO+K-mean	2016	[53]
11	Firefly Algorithm	GF-CLUST	2016	[95]
12	Hierarchical Clustering Algorithm	Hier	2016	[96]
13	K-mean clustering with PSO feature selection	H-FSPSOTC	2017	[4]
14	Link based particle swarm optimization	LBPSO	2017	[97]
15	Dimension reduction and feature section for K-mean	(LFW)PSO-DDR	2017	[47]
16	A combination method of GSA and K-harmonic means	GSA-KHM	2017	[98]
17	Hybrid Improved Krill Herd Algorithm	MMKHA	2018	[65]
18	Combination of Objective Functions and HKHA	MHKHA	2018	[1]
19	Gray Wolf Optimizer	TCP-GWO	2019	[19]
20	β-hill climbing	βHC	2020	[54]
21	K-means clustering	K-means	2020	[33]
22	Improved Arithmetic Optimization Algorithm	IAOA	2021	The proposed

according to the evaluation measure in the underlying dataset. The best results of all evaluation measures are the highest; only the best entropy result is the lowest. The proposed IAOA got the best results in precision measures in eight out of eleven comparisons. The proposed IAOA got the best results in ten out of eleven comparisons in the recall measure. According to the F-measure results, the proposed IAOA got the best results in twenty-six comparisons. The proposed IAOA got the best results in nineteen out of fourteen comparisons according to the Purity measure. Finally, it got the best results in fourteen out of twenty-two comparisons, as shown in Fig. 7. The results obtained by the proposed IAOA overwhelm almost all the comparative state-of-the-art methods published by several well-known researchers who have performed significant work to get the best clustering outcomes.

From the previous results and discussion, the developed IAOA method's high ability to find a suitable solution to various optimization problems and data clustering. By comparing the performance of IAOA with the traditional AOA method, it can be seen the enormous effect of Levy Flight and OBL to enhance the ability of balance between the exploration and exploitation observed in the high quality of the final solution. However, the developed method still needs some improvement, especially in time computation, since it depends on OBL to increase the processing time.

5. Conclusion

With the massive increase of WEB data, social media, and smart devices, text document clustering has become an increasingly critical task. This paper proposed a novel text document clustering approach by adopting a modified version of the recently proposed optimization algorithm, called Arithmetic Optimization Algorithm (AOA). We improved AOA by applying two search mechanisms, Opposition-based learning (OBL) and Levy Flight Distribution (LFD), to enhance its searchability and avoid its original version's shortcomings (i.e., premature convergence, stuck in local optima), called IAOA.

The proposed IAOA is evaluated using different LABIC clustering datasets for text clustering. It has been compared to several existing approaches. The obtained results showed that the proposed IAOA is a superior method in solving the text clustering problem and better than the other comparative methods

Table 18
Description of text datasets used by the state-of-the-art methods.

Dataset	Source	# of documents	# of clusters	Reference
C-DS01	Reuters-21,578	1339	08	[50] [11] [53]
C-DS02	Classic4	2000	04	[50] [11] [53]
C-DS03	WebKB	2803	04	[50] [11] [53]
C-DS04	20Newsgroups	2000	11	[91]
C-DS05	Reuters-21587	1000	05	[91] [98]
C-DS06	20Newsgroups	300	03	[95] [12]
C-DS07	Reuters-21578	300	06	[95]
C-DS08	TREC	414	09	[93] [95] [93] [47] [97] [54] [4] [33]
C-DS09	TREC	313	08	[93] [47]
C-DS10	20Newsgroups	300	03	[12]
C-DS11	WebACE	300	03	[92]
C-DS12	Reuters-21587	1049	10	[90]
C-DS13	Classic4	500	04	[90]
C-DS14	Classic4	1000	04	[90]
C-DS15	20Newsgroups	500	20	[90]
C-DS16	20Newsgroups	1000	20	[90]
C-DS17	TREC	204	06	[93] [47] [54] [4] [33]
C-DS18	TREC	927	07	[93]
C-DS19	TREC	690	10	[93]
C-DS20	Reuters-21587	200	10	[93]
C-DS21	20Newsgroups	2000	20	[93] [98]
C-DS22	Reuters-21587	180	06	[96]
C-DS23	TREC	873	08	[62] [94]
C-DS24	DMOZ	697	14	[62] [94]
C-DS25	20Newsgroups	9249	10	[62] [94]
C-DS26	WebAce	1560	20	[62] [94]
C-DS27	Technical reports	299	4	[47] [54] [4] [33]
C-DS28	Web pages	334	4	[47] [54] [4]
C-DS29	TREC	878	10	[47] [33]
C-DS30	MEDLINE	913	10	[47] [33]
C-DS31	20newsgroup	18828	20	[47]

Table 19
The results obtained by the proposed IAOA and best-published results.

Dataset	Method	Precision	Recall	F-measure	Purity	Entropy
C-DS01	22	0.4958	**0.5012**	**0.4984**	–	–
	9	0.5392	0.3224	0.4022	–	–
	10	0.5841	0.3760	0.4550	–	–
	8	**0.6752**	0.3790	0.4855	–	–
C-DS02	22	0.7468	0.7254	0.7359	–	–
	9	**0.8881**	**0.8254**	0.8080	–	–
	10	0.6161	–	0.6716	–	–
	8	0.8798	0.8065	**0.8416**	–	–
C-DS03	22	0.4102	**0.4028**	**0.4064**	–	–
	9	0.4562	0.3503	0.3948	–	–
	10	0.3926	0.3668	0.3820	–	–
	8	**0.4726**	0.3340	0.3914	–	–
C-DS04	22			0.4325	–	0.3562
	3	–	–	**0.4800**	–	**0.3400**
C-DS05	22	–	–	**0.4325**	–	0.4745
	3	–	–	0.3200	–	0.4700
	1	–	–	–	–	0.5350
	16	–	–	0.3400	–	**0.4600**
C-DS06	22	–	–	**0.5743**	**0.5821**	**0.4982**
	11	–	–	0.5218	0.5667	1.3172
C-DS07	22	–	–	**0.4032**	**0.5514**	**0.5147**
	11	–	–	0.3699	0.4867	1.6392
C-DS08	22	**0.4952**	**0.5101**	**0.5025**	0.6525	**0.4325**
	11	–	–	0.3213	0.4710	2.0119
	5	–	–	0.2478	0.4850	1.4102
	15	0.4824	0.4649	0.4727	**0.8406**	0.4419
	20	0.3014	0.2951	0.3054	0.4510	1.2540
	13	0.3515	0.3321	0.3415	0.5410	0.9541
	14	–	–	–	0.7319	–
	21	0.4486	0.43362	0.4405	0.6838	0.46696
C-DS09	22	**0.5214**	**0.5102**	**0.5157**	**0.7963**	0.6025
	11	–	–	0.3851	0.4920	1.1246
	5	–	–	0.1946	0.3514	1.7344
	15	0.5176	0.5193	0.5143	0.7864	**0.5277**
C-DS10	22	–	–	0.5165	0.7325	**0.6654**
	7	–	–	**0.5753**	**0.7655**	0.8118
C-DS11	22	–	–	**0.7254**	**0.7025**	0.7021
	4	–	–	0.7110	0.6910	**0.6730**
C-DS12	22	–	–	–	**0.6325**	–
	1	–	–	–	0.4000	–
C-DS13	22	–	–	–	**0.5548**	–
	1	–	–	–	0.4760	–
C-DS14	22	–	–	–	**0.5254**	–
	1	–	–	–	0.5130	–
C-DS15	22	–	–	–	**0.4040**	–
	1	–	–	–	0.3970	–
C-DS16	22	–	–	–	**0.4369**	–
	1	–	–	–	0.4000	–
C-DS17	22	**0.4125**	**0.4032**	**0.4078**	0.6474	0.3452
	5	–	–	0.1719	0.4853	1.3351
	15	0.3617	0.3587	0.4059	**0.6601**	0.4616
	20	0.2741	0.2961	0.2845	0.3451	1.1410
	13	0.3369	0.3514	0.3499	0.4951	0.7562
	21	0.3690	0.3906	0.3772	0.6321	**0.3339**
C-DS18	22	–	–	**0.3365**	–	**0.4012**
	5	–	–	0.1407	–	0.4344
C-DS19	22	–	–	**0.2982**	0.4102	**0.7585**
	5	–	–	0.2627	**0.4210**	1.5922
C-DS20	22	–	–	**0.3025**	**0.2758**	**0.4858**
	5	–	–	0.2444	0.2518	1.9981
C-DS21	22	–	–	0.3562	**0.4555**	0.7584
	5	–	–	0.1894	0.2141	2.2575
	16	–	–	**0.5100**	–	**0.3400**

(*continued on next page*)

Table 19 (*continued*).

Dataset	Method	Precision	Recall	F-measure	Purity	Entropy
C-DS22	22	–	–	**0.4586**	–	–
	12	–	–	0.2800	–	–
C-DS23	22	–	–	0.80147	0.7054	**0.4252**
	2	–	–	**0.8968**	**0.7611**	0.4401
	6	–	–	0.8826	–	–
C-DS24	22	–	–	**0.8544**	**0.7456**	0.3458
	2	–	–	0.8397	0.7328	**0.3312**
	6	–	–	0.8574	–	–
C-DS25	22	–	–	0.7802	**0.6800**	**0.5056**
	2	–	–	0.7805	0.6712	0.5083
	6	–	–	**0.7902**	–	–
C-DS26	22	–	–	**0.7750**	**0.5741**	**0.4659**
	2	–	–	0.7662	0.5647	0.4701
	6	–	–	0.7342	–	–
C-DS27	22	**0.6014**	**0.6214**	**0.6112**	**0.7951**	0.4169
	15	0.5127	0.5203	0.5648	0.6418	**0.4088**
	20	0.2541	0.3047	0.2854	0.4125	0.8954
	13	0.3515	0.3547	0.3524	0.5146	0.7841
	16	0.5966	0.5998	0.5994	–	–
	21	0.59883	0.61621	0.60667	0.77867	0.33315
C-DS28	22	**0.7552**	**0.7841**	**0.7694**	**0.9991**	**0.3891**
	15	0.7512	0.7721	0.7611	0.9902	0.3913
	20	0.5314	0.5433	0.5454	0.7051	0.6541
	13	0.5800	0.5984	0.5840	0.8100	0.4650
	19	0.7500	0.6873	0.7156	–	–
C-DS29	22	**0.5236**	**0.531**	**0.5273**	**0.6925**	0.3521
	15	0.4883	0.5254	0.5054	0.6783	**0.3313**
	20	0.2221	0.2104	0.2185	–	–
	21	0.45106	0.48576	0.46706	0.65659	0.38764
C-DS30	22	**0.5123**	**0.529**	**0.5205**	**0.6852**	**0.5986**
	15	0.4991	0.4837	0.4830	0.6386	0.6662
	20	0.2514	0.2541	0.2541	–	–
	21	0.494360	0.499522	0.496694	0.663414	0.624934
C-DS31	22	**0.4250**	**0.4139**	**0.4194**	**0.5325**	**0.5552**
	15	0.3956	0.4099	0.4011	0.4958	0.5983
Best results	22	8/11	10/11	20/26	19/24	14/22

Note: The best results are highlighted in **bold** font.
Note: The highest value is the best one except the Entropy measure, the lowest values is the best one.

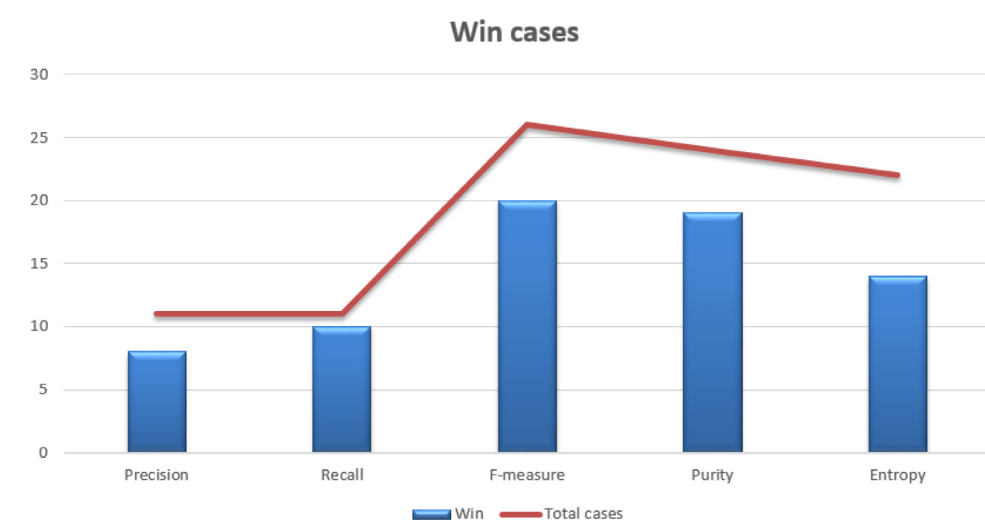

Fig. 7. Win cases obtained by the proposed IAOA.

published in the literature. Moreover, we also considered the proposed IAOA with CEC2019 benchmark functions to test its performance as a global optimizer. All experiments confirmed the competitive performance of the IAOA compared to other optimization algorithms. This proved that the incorporated methods (i.e., OBL and LFD) into the conventional AOA effectively improved the searchability of AOA and kept the solutions diverse. The equilibrium between the exploration and exploitation search methods is also accomplished in the proposed IAOA.

With the superior performance of the IAOA, in the future, it may be further tested to be applied for other optimization tasks, for example, data clustering, medical image segmentation, time-series prediction, cloud computing scheduling, text classification, text summarizing, engineering design problems, feature selection, multi-objective optimization problems, parameter extraction of photovoltaic, other numerical optimization problems and others. Moreover, the proposed method can be further improved using other optimization procedures, such as a novel local search technique, genetic operations, multi-population strategy, fractional-order calculus, k-mean clustering technique, etc. The main limitations of the proposed method have risen as the used benchmark can be replaced with other benchmark functions, and a real word application is needed.

CRediT authorship contribution statement

Laith Abualigah: Supervision, Conceptualization, Methodology, Software, Investigation, Validation, Writing – original draft preparation. **Khaled H. Almotairi:** Writing – original draft preparation, Visualization, Investigation. **Mohammed A.A. Al-qaness:** Writing – original draft preparation, Visualization, Investigation. **Ahmed A. Ewees:** Writing – original draft preparation, Visualization, Investigation. **Dalia Yousri:** Writing – original draft preparation, Visualization, Investigation. **Mohamed Abd Elaziz:** Writing – original draft preparation, Visualization, Investigation. **Mohammad H. Nadimi-Shahraki:** Writing – original draft preparation, Visualization, Investigation.

Declaration of competing interest

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

Acknowledgments

The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University, Saudi Arabia for supporting this work by Grant Code: (22UQU4320277DSR05).

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent

Informed consent was obtained from all individual participants included in the study.

References

[1] L.M. Abualigah, A.T. Khader, E.S. Hanandeh, A combination of objective functions and hybrid krill herd algorithm for text document clustering analysis, Eng. Appl. Artif. Intell. 73 (2018) 111–125.
[2] N. Kushwaha, M. Pant, Link based BPSO for feature selection in big data text clustering, Future Gener. Comput. Syst. 82 (2018) 190–199.
[3] R. Janani, S. Vijayarani, Text document clustering using spectral clustering algorithm with particle swarm optimization, Expert Syst. Appl. 134 (2019) 192–200.
[4] L.M. Abualigah, A.T. Khader, Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering, J. Supercomput. 73 (11) (2017) 4773–4795.
[5] A.H. Gandomi, F. Chen, L. Abualigah, Machine learning technologies for big data analytics, Electronics 11 (3) (2022) 421.
[6] F. Zabihi, B. Nasiri, A novel history-driven artificial bee colony algorithm for data clustering, Appl. Soft Comput. 71 (2018) 226–241.
[7] C. Liu, Y. Li, Q. Zhao, C. Liu, Reference vector-based multi-objective clustering for high-dimensional data, Appl. Soft Comput. 78 (2019) 614–629.
[8] A. Santra, C.J. Christy, Genetic algorithm and confusion matrix for document clustering, Int. J. Comput. Sci. Issues (IJCSI) 9 (1) (2012) 322.
[9] B. Abdollahzadeh, F.S. Gharehchopogh, S. Mirjalili, African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems, Comput. Ind. Eng. 158 (2021) 107408.
[10] M.J. Goldanloo, F.S. Gharehchopogh, A hybrid OBL-based firefly algorithm with symbiotic organisms search algorithm for solving continuous optimization problems, J. Supercomput. 78 (3) (2022) 3998–4031.
[11] K.K. Bharti, P.K. Singh, Chaotic gradient artificial bee colony for text clustering, Soft Comput. 20 (3) (2016) 1113–1126.
[12] A.J. Mohammed, Y. Yusof, H. Husni, Document clustering based on firefly algorithm, J. Comput. Sci. 11 (3) (2015) 453.
[13] J.O. Agushaka, A.E. Ezugwu, L. Abualigah, Dwarf mongoose optimization algorithm, Comput. Methods Appl. Mech. Engrg. 391 (2022) 114570.
[14] L.M.Q. Abualigah, Feature Selection and Enhanced Krill Herd Algorithm for Text Document Clustering, Springer, 2019.
[15] H. Yu, Z. Chang, G. Wang, X. Chen, An efficient three-way clustering algorithm based on gravitational search, Int. J. Mach. Learn. Cybern. 11 (5) (2020) 1003–1016.
[16] A. Abraham, S. Das, A. Konar, Document clustering using differential evolution, in: 2006 IEEE International Conference on Evolutionary Computation, IEEE, 2006, pp. 1784–1791.
[17] H. Shayanfar, F.S. Gharehchopogh, Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems, Appl. Soft Comput. 71 (2018) 728–746.
[18] F.S. Gharehchopogh, B. Abdollahzadeh, An efficient harris hawk optimization algorithm for solving the travelling salesman problem, Cluster Comput. (2021) 1–25.
[19] H. Rashaideh, A. Sawaie, M.A. Al-Betar, L.M. Abualigah, M.M. Al-Laham, M. Ra'ed, M. Braik, A grey wolf optimizer for text document clustering, J. Intell. Syst. 29 (1) (2018) 814–830.
[20] M.H. Nadimi-Shahraki, A. Fatahi, H. Zamani, S. Mirjalili, L. Abualigah, M. Abd Elaziz, Migration-based moth-flame optimization algorithm, Processes 9 (12) (2021) 2276.
[21] H. Zamani, M.H. Nadimi-Shahraki, A.H. Gandomi, Starling murmuration optimizer: A novel bio-inspired algorithm for global and engineering optimization, Comput. Methods Appl. Mech. Engrg. 392 (2022) 114616.
[22] S.I. Boushaki, N. Kamel, O. Bendjeghaba, Improved cuckoo search algorithm for document clustering, in: IFIP International Conference on Computer Science and Its Applications, Springer, 2015, pp. 217–228.
[23] A.K. Abasi, A.T. Khader, M.A. Al-Betar, S. Naim, S.N. Makhadmeh, Z.A.A. Alyasseri, Link-based multi-verse optimizer for text documents clustering, Appl. Soft Comput. 87 (2020) 106002.
[24] O.N. Oyelade, A.E.-S. Ezugwu, T.I. Mohamed, L. Abualigah, Ebola optimization search algorithm: A new nature-inspired metaheuristic optimization algorithm, IEEE Access 10 (2022) 16150–16177.
[25] M. Belazzoug, M. Touahria, F. Nouioua, M. Brahimi, An improved sine cosine algorithm to select features for text categorization, J. King Saud Univ.-Comput. Inf. Sci. 32 (4) (2020) 454–464.
[26] F.S. Gharehchopogh, Advances in tree seed algorithm: A comprehensive survey, Arch. Comput. Methods Eng. (2022) 1–24.
[27] H.R.R. Zaman, F.S. Gharehchopogh, An improved particle swarm optimization with backtracking search optimization algorithm for solving continuous optimization problems, Eng. Comput. (2021) 1–35.
[28] L. Abualigah, D. Yousri, M. Abd Elaziz, A.A. Ewees, M.A. Al-Qaness, A.H. Gandomi, Aquila optimizer: a novel meta-heuristic optimization algorithm, Comput. Ind. Eng. 157 (2021) 107250.
[29] S. Ghafori, F.S. Gharehchopogh, Advances in spotted hyena optimizer: a comprehensive survey, Arch. Comput. Methods Eng. (2021) 1–22.
[30] V. Kumaresan, R. Nagarajan, Design and analysis of text document clustering using moth-flame optimization, J. Crit. Rev. 7 (18) (2020) 792–794.
[31] H. Mohmmadzadeh, F.S. Gharehchopogh, An efficient binary chaotic symbiotic organisms search algorithm approaches for feature selection problems, J. Supercomput. 77 (8) (2021) 9102–9144.
[32] L. Abualigah, M. Abd Elaziz, P. Sumari, Z.W. Geem, A.H. Gandomi, Reptile search algorithm (RSA): A nature-inspired meta-heuristic optimizer, Expert Syst. Appl. 191 (2022) 116158.
[33] L. Abualigah, A.H. Gandomi, M.A. Elaziz, A.G. Hussien, A.M. Khasawneh, M. Alshinwan, E.H. Houssein, Nature-inspired optimization algorithms for text document clustering—A comprehensive analysis, Algorithms 13 (12) (2020) 345.
[34] M.H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, A.A. Ewees, L. Abualigah, M. Abd Elaziz, MTV-MFO: Multi-trial vector-based moth-flame optimization algorithm, Symmetry 13 (12) (2021) 2388.
[35] M.H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, L. Abualigah, M. Abd Elaziz, D. Oliva, EWOA-OPF: Effective whale optimization algorithm to solve optimal power flow problem, Electronics 10 (23) (2021) 2975.
[36] L. Abualigah, A.H. Gandomi, M.A. Elaziz, H.A. Hamad, M. Omari, M. Alshinwan, A.M. Khasawneh, Advances in meta-heuristic optimization algorithms in big data text clustering, Electronics 10 (2) (2021) 101.

[37] L. Abualigah, A. Diabat, Z.W. Geem, A comprehensive survey of the harmony search algorithm in clustering applications, Appl. Sci. 10 (11) (2020) 3827.
[38] R. Purushothaman, S. Rajagopalan, G. Dhandapani, Hybridizing gray wolf optimization (GWO) with Grasshopper optimization algorithm (GOA) for text feature selection and clustering, Appl. Soft Comput. 96 (2020) 106651.
[39] I. Aljarah, M. Mafarja, A.A. Heidari, H. Faris, S. Mirjalili, Clustering analysis using a novel locality-informed grey wolf-inspired clustering approach, Knowl. Inf. Syst. 62 (2) (2020) 507–539.
[40] T. Bezdan, C. Stoean, A.A. Naamany, N. Bacanin, T.A. Rashid, M. Zivkovic, K. Venkatachalam, Hybrid fruit-fly optimization algorithm with k-means for text document clustering, Mathematics 9 (16) (2021) 1929.
[41] G. Costa, R. Ortale, Effective interrelation of Bayesian nonparametric document clustering and embedded-topic modeling, Knowl.-Based Syst. 234 (2021) 107591.
[42] N. Rahnema, F.S. Gharehchopogh, An improved artificial bee colony algorithm based on whale optimization algorithm for data clustering, Multimedia Tools Appl. 79 (43) (2020) 32169–32194.
[43] J.M. Sanchez-Gomez, M.A. Vega-Rodríguez, C.J. Pérez, A decomposition-based multi-objective optimization approach for extractive multi-document text summarization, Appl. Soft Comput. (2020) 106231.
[44] M.A. Mosa, A novel hybrid particle swarm optimization and gravitational search algorithm for multi-objective optimization of text mining, Appl. Soft Comput. 90 (2020) 106189.
[45] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A.H. Gandomi, The arithmetic optimization algorithm, Comput. Methods Appl. Mech. Engrg. 376 (2020) 113609.
[46] H.R. Tizhoosh, Opposition-based learning: a new scheme for machine intelligence, in: International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), vol. 1, IEEE, 2005, pp. 695–701.
[47] L.M. Abualigah, A.T. Khader, M.A. Al-Betar, O.A. Alomari, Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering, Expert Syst. Appl. 84 (2017) 24–36.
[48] L.M. Abualigah, A.T. Khader, E.S. Hanandeh, A novel weighting scheme applied to improve the text document clustering techniques, in: Innovative Computing, Optimization and Its Applications, Springer, 2018, pp. 305–320.
[49] A.E. Ezugwu, A.M. Ikotun, O.O. Oyelade, L. Abualigah, J.O. Agushaka, C.I. Eke, A.A. Akinyelu, A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects, Eng. Appl. Artif. Intell. 110 (2022) 104743.
[50] K.K. Bharti, P.K. Singh, Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering, Expert Syst. Appl. 42 (6) (2015) 3105–3114.
[51] L.M. Abualigah, A.T. Khader, E.S. Hanandeh, A.H. Gandomi, A novel hybridization strategy for krill herd algorithm applied to clustering techniques, Appl. Soft Comput. 60 (2017) 423–435.
[52] K.K. Bharti, P.K. Singh, A three-stage unsupervised dimension reduction method for text clustering, J. Comput. Sci. 5 (2) (2014) 156–169.
[53] K.K. Bharti, P.K. Singh, Opposition chaotic fitness mutation based adaptive inertia weight BPSO for feature selection in text clustering, Appl. Soft Comput. (2016).
[54] L.M. Abualigah, E.S. Hanandeh, A.T. Khader, M.A. Otair, S.K. Shandilya, An improved b-hill climbing optimization technique for solving the text documents clustering problem, Curr. Med. Imaging 16 (4) (2020) 296–306.
[55] G. Salton, A. Wong, C.-S. Yang, A vector space model for automatic indexing, Commun. ACM 18 (11) (1975) 613–620.
[56] J.M. Rohani, S.M. Zahraee, Production line analysis via value stream mapping: a lean manufacturing process of color industry, Procedia Manuf. 2 (2015) 6–10.
[57] L.M.Q. Abualigah, E.S. Hanandeh, Applying genetic algorithms to information retrieval using vector space model, Int. J. Comput. Sci. Eng. Appl. 5 (1) (2015) 19.
[58] J. Yun, L. Jing, J. Yu, H. Huang, A multi-layer text classification framework based on two-level representation model, Expert Syst. Appl. 39 (2) (2012) 2035–2046.
[59] L.-P. Jing, H.-K. Huang, H.-B. Shi, Improved feature selection approach TFIDF in text mining, in: Proceedings. International Conference on Machine Learning and Cybernetics, vol. 2, IEEE, 2002, pp. 944–946.
[60] J.A. Nasir, I. Varlamis, A. Karim, G. Tsatsaronis, Semantic smoothing for text clustering, Knowl.-Based Syst. 54 (2013) 216–229.
[61] M. Mahdavi, H. Abolhassani, Harmony K-means algorithm for document clustering, Data Min. Knowl. Discov. 18 (3) (2009) 370–391.
[62] R. Forsati, M. Mahdavi, M. Shamsfard, M.R. Meybodi, Efficient stochastic algorithms for document clustering, Inform. Sci. 220 (2013) 269–291.
[63] L.M. Abualigah, A.T. Khader, E.S. Hanandeh, A new feature selection method to improve the document clustering using particle swarm optimization algorithm, J. Comput. Sci. 25 (2018) 456–466.
[64] A. Moayedikia, R. Jensen, U.K. Wiil, R. Forsati, Weighted bee colony algorithm for discrete optimization problems with application to feature selection, Eng. Appl. Artif. Intell. 44 (2015) 153–167.
[65] L.M. Abualigah, A.T. Khader, E.S. Hanandeh, Hybrid clustering analysis using improved krill herd algorithm, Appl. Intell. 48 (11) (2018) 4047–4071.
[66] R. Forsati, M. Mahdavi, Web text mining using harmony search, in: Recent Advances in Harmony Search Algorithm, Springer, 2010, pp. 51–64.
[67] K.H. Almotairi, L. Abualigah, Hybrid reptile search algorithm and remora optimization algorithm for optimization tasks and data clustering, Symmetry 14 (3) (2022) 458.
[68] A.A. Ewees, M. Abd Elaziz, E.H. Houssein, Improved grasshopper optimization algorithm using opposition-based learning, Expert Syst. Appl. 112 (2018) 156–172.
[69] Q. Xu, L. Wang, N. Wang, X. Hei, L. Zhao, A review of opposition-based learning from 2005 to 2012, Eng. Appl. Artif. Intell. 29 (2014) 1–12.
[70] S. Gupta, K. Deep, A hybrid self-adaptive sine cosine algorithm with opposition based learning, Expert Syst. Appl. 119 (2019) 210–230.
[71] S.N. Chegini, A. Bagheri, F. Najafi, PSOSCALF: A new hybrid PSO based on Sine cosine algorithm and levy flight for solving optimization problems, Appl. Soft Comput. 73 (2018) 697–726.
[72] R.A. Ibrahim, M. Abd Elaziz, S. Lu, Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization, Expert Syst. Appl. 108 (2018) 1–27.
[73] M. Abd Elaziz, A.A. Heidari, H. Fujita, H. Moayedi, A competitive chain-based Harris Hawks Optimizer for global optimization and multi-level image thresholding problems, Appl. Soft Comput. (2020) 106347.
[74] A.H. Elsheikh, T.A. Shehabeldeen, J. Zhou, E. Showaib, M. Abd Elaziz, Prediction of laser cutting parameters for polymethylmethacrylate sheets using random vector functional link network integrated with equilibrium optimizer, J. Intell. Manuf. (2020) 1–12.
[75] M.A. Al-Qaness, A.A. Ewees, H. Fan, L. Abualigah, M. Abd Elaziz, Marine predators algorithm for forecasting confirmed cases of COVID-19 in Italy, USA, Iran and Korea, Int. J. Environ. Res. Public Health 17 (10) (2020) 3520.
[76] M. Abd Elaziz, A.A. Ewees, D. Yousri, H.S.N. Alwerfali, Q.A. Awad, S. Lu, M.A. Al-Qaness, An improved marine predators algorithm with fuzzy entropy for multi-level thresholding: Real world example of COVID-19 CT image segmentation, IEEE Access 8 (2020) 125306–125330.
[77] A.A. Ewees, M. Abd Elaziz, Performance analysis of chaotic multi-verse harris hawks optimization: a case study on solving engineering problems, Eng. Appl. Artif. Intell. 88 (2020) 103370.
[78] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, 4, IEEE, 1995, pp. 1942–1948.
[79] H. Jouhari, D. Lei, M. A.A. Al-qaness, M. Abd Elaziz, A.A. Ewees, O. Farouk, Sine-cosine algorithm to enhance simulated annealing for unrelated parallel machine scheduling with setup times, Mathematics 7 (11) (2019) 1120.
[80] R.A. Ibrahim, A.A. Ewees, D. Oliva, M. Abd Elaziz, S. Lu, Improved salp swarm algorithm based on particle swarm optimization for feature selection, J. Ambient Intell. Humaniz. Comput. 10 (8) (2019) 3155–3169.
[81] N. Neggaz, A.A. Ewees, M. Abd Elaziz, M. Mafarja, Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection, Expert Syst. Appl. 145 (2020) 113103.
[82] M. Abd Elaziz, S. Mirjalili, A hyper-heuristic for improving the initial population of whale optimization algorithm, Knowl.-Based Syst. 172 (2019) 42–63.
[83] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, Equilibrium optimizer: A novel optimization algorithm, Knowl.-Based Syst. 191 (2020) 105190.
[84] M.S. Nobile, P. Cazzaniga, D. Besozzi, R. Colombo, G. Mauri, G. Pasi, Fuzzy self-tuning PSO: A settings-free algorithm for global optimization, Swarm Evol. Comput. 39 (2018) 70–85.
[85] G.-G. Wang, M. Lu, X.-J. Zhao, An improved bat algorithm with variable neighborhood search for global optimization, in: 2016 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2016, pp. 1773–1778.
[86] H. Zhang, M. Yuan, Y. Liang, Q. Liao, A novel particle swarm optimization based on prey–predator relationship, Appl. Soft Comput. 68 (2018) 202–218.
[87] G.-G. Wang, A.H. Gandomi, A.H. Alavi, G.-S. Hao, Hybrid krill herd algorithm with differential evolution for global numerical optimization, Neural Comput. Appl. 25 (2) (2014) 297–308.
[88] G.-G. Wang, S. Deb, A.H. Gandomi, Z. Zhang, A.H. Alavi, Chaotic cuckoo search, Soft Comput. 20 (9) (2016) 3349–3362.
[89] G.-G. Wang, A.H. Gandomi, A.H. Alavi, Stud krill herd algorithm, Neurocomputing 128 (2014) 363–370.
[90] V.K. Singh, N. Tiwari, S. Garg, Document clustering using k-means, heuristic k-means and fuzzy c-means, in: Computational Intelligence and Communication Networks (CICN), 2011 International Conference on, IEEE, 2011, pp. 297–301.
[91] S. Karol, V. Mangat, Evaluation of text document clustering approach based on particle swarm optimization, Open Comput. Sci. 3 (2) (2013) 69–90.

[92] M.M. Zaw, E.E. Mon, Web document clustering using cuckoo search clustering algorithm based on levy flight, Int. J. Innov. Appl. Stud. 4 (1) (2013) 182–188.
[93] B. Prakash, M. Hanumanthappa, M. Mamatha, Cluster based term weighting model for web document clustering, in: Proceedings of the Third International Conference on Soft Computing for Problem Solving, Springer, 2014, pp. 815–822.
[94] R. Forsati, A. Keikha, M. Shamsfard, An improved bee colony optimization algorithm with an application to document clustering, Neurocomputing 159 (2015) 9–26.
[95] A.J. Mohammed, Y. Yusof, H. Husni, GF-CLUST: A nature-inspired algorithm for automatic text clustering, J. Inf. Commun. Technol. 15 (1) (2016).
[96] J.D. Rose, An efficient association rule based hierarchical algorithm for text clustering, Int. J. Adv. Eng. Tech./Vol. VII/Issue I/Jan.-March 751 (2016) 753.
[97] N. Kushwaha, M. Pant, Link based BPSO for feature selection in big data text clustering, Future Gener. Comput. Syst. (2017).
[98] M. Mirhosseini, A clustering approach using a combination of gravitational search algorithm and k-harmonic means and its application in text document clustering, Turk. J. Electr. Eng. Comput. Sci. 25 (2) (2017) 1251–1262.