ITC 2/52
Information Technology and Control
Vol. 52 / No. 2 / 2023
pp. 541-561
DOI 10.5755/j01.itc.52.2.33536

Text Document Clustering Approach by Improved Sine Cosine Algorithm

Received 2023/03/01 | Accepted after revision 2023/04/14

HOW TO CITE: Radomirović, B., Jovanović, V., Nikolić, B., Stojanović, S., Venkatachalam, K., Zivkovic, M., Njeguš, A., Bacanin, N., Strumberger, I. (2023). Text Document Clustering Approach by Improved Sine Cosine Algorithm. *Information Technology and Control,* 52(2), 541-561. https://doi.org/10.5755/j01.itc.52.2.33536

Text Document Clustering Approach by Improved Sine Cosine Algorithm

Branislav Radomirović

Singidunum University Danijelova 32, 11000 Belgrade, Serbia

Vuk Jovanović, Bosko Nikolić, Sasa Stojanović

School of Electrical Engineering, University of Belgrade Bul. kralja Aleksandra 73, 11000 Belgrade, Serbia

K. Venkatachalam

University of Hradec Kralove 50003 Hradec Kralove, Czech Republic

Miodrag Zivkovic, Angelina Njeguš

Singidunum University Danijelova 32, 11000 Belgrade, Serbia

Nebojsa Bacanin

Singidunum University Danijelova 32, 11000 Belgrade, Serbia; MEU Research Unit, Middle East University, Amman, Jordan

Ivana Strumberger

Singidunum University Danijelova 32, 11000 Belgrade, Serbia

Corresponding author: venkatachalam.k@ieee.org

Due to the vast amounts of textual data available in various forms such as online content, social media comments, corporate data, public e-services and media data, text clustering has been experiencing rapid development. Text clustering involves categorizing and grouping similar content. It is a process of identifying significant patterns from unstructured textual data. Algorithms are being developed globally to extract useful and relevant information from large amounts of text data. Measuring the significance of content in documents to partition the collection of text data is one of the most important obstacles in text clustering. This study suggests utilizing an improved metaheuristics algorithm to fine-tune the K-means approach for text clustering task. The suggested technique is evaluated using the first 30 unconstrained test functions from the CEC2017 test-suite and six standard criterion text datasets. The simulation results and comparison with existing techniques demonstrate the robustness and supremacy of the suggested method.

KEYWORDS: text document clustering, optimization problems, metaheuristics, sine cosine algorithm, hybridization and K-means.

1. Introduction

The amount of text stored daily is growing exponentially, therefore, in recent decades a methodology has developed for the selection and processing of a vast volume of unstructured textual data. Today's computers efficiently apply techniques and algorithms that process textual data. Textual data is considered unstructured data and as such requires a special form of access, storage, processing, and presentation. Textual data is the basis for the application of text mining analysis, which is just one of the challenges of the machine and cognitive learning. Text mining is a data mining technique that aims to convert large volumes of text into useful and comprehensible information by selecting appropriate sources extracting templates, and transforming the text. Clustering is a frequently used text mining procedure that involves grouping data into a collection of clusters, typically with a fixed count of clusters, based on similarities among the elements. Text clustering is a crucial method in text mining as it involves classifying and grouping similar content. Based on content, all documents in one cluster are related to one other, however, in other clusters, the similarity declines. Clustering is an NP-hard challenge in optimization that cannot be solved through employment of the traditional deterministic techniques.

Metaheuristics algorithms are stochastic methods that have proven to be particularly efficient at resolving NP-hard challenges in optimization and producing almost optimal solution in an acceptable lapse of time. Regarding the classification of the metaheuristics, the biggest group that has been motivated by nature may be split to two main classes, namely swarm intelligence and evolutionary algorithms. Another recent group of metaheuristics algorithms uses the fundamental mathematical laws for guiding the search, where methods like sine cosine algorithm belong to. This work suggests an improved metaheuristics method by using the sine cosine algorithm and improving it with additional operators from the genetic algorithm.

Image segmentation, classification, clustering (Zivkovic et al. [48], Sarac et al. [38], Bacanin et al. [6]), feature selection (Bezdan et al. [9]), convolutional neural network architecture optimization (Strumberger et al. [42], Bukumira et al. [12]), cloud computing applications (Bacanin et al. [7], Zivkovic et al. [50]), intrusion detection and credit card malverstions (Zivkovic et al. [51], Jovanovic et al. [25]), and time-series forecasting (Jovanovic et al. [27], Stoean et al. [40], Zivkovic et al. [49], Bacanin et al. [5], Jovanovic et al. [26]) are just a few of the many successful applications of metaheuristics methods in general.

The introduced approach uses the K-means algorithm, and its main objective is the updating of the centroid of the cluster, where data points are defined as centers. The calculation is repeated until it converges. After introducing a unique technique for term weighting based on distance to encode them by taking into account distances in between the news terms and if terms have appeared. Chen [14] used K-means amongst other clustering methods. The proposed research has the potential to significantly increase clustering performance.

The main goal of this manuscript was to provide the answers to the following research questions:

- Develop an enhanced sine cosine method that outperforms the original algorithm by the solution quality
- Use the new method to tune K-means for text document clustering and achieve superior results than previous approaches for the same problem

The rest of the document is structured in the following way. The background and related work is defined in Section 2, the text document clustering problem statement is defined in Section 3 and the description of the suggested method is presented in Section 4, Section 5 brings forward the simulation outcome of the experiments executed on CEC 2017 unconstrained benchmark functions. Finally, Section 6 completes the paper.

2. Background and Related Work

Text document clustering is a significant machine learning task, and algorithms based on metaheuristics have been frequently utilized to solve it. For example in, Chen et al. [15] the particle swarm optimization algorithm (PSO) was hybridized with the K-means method and OTSU algorithm for text clustering. In Purushothaman et al. [35], a hybrid approach was ap-

plied for feature selection and text clustering by combining the grey wolf optimization algorithm (GWO) with the grasshopper optimization algorithm (GOA) and Fuzzy c-means (FCM). Another hybrid approach for web text document clustering was proposed in Abualigah et al. [2] which combined the krill herd algorithm (KH) with genetic operators. Additionally, Abasi et al. [1] applied a hybrid multi-verse optimizer (MVO) in conjunction with the traditional K-means clustering algorithm for text clustering. The genetic algorithm utilizing the ontology strategy was used for text clustering in Song et al. [39] while the hybrid artificial bee colony (ABC) algorithm was applied to enhance data clustering in Karaboga and Ozturk [29]. Additionaly the social spider optimization (SSO) path was utilized for clustering text documents in Chandran et al. [13]. Mor recent studies have explored the use of the sine cosine algorithm (SCA in Mirijalili [31], salp swarm algorithm (SSA) in Ponnusamy et al. [33], and firefly algorithm (FA) in Tomer and Kumar [44]. Despite the existence of numerous metaheuristic-based text clustering algorithms, it is always necessary to create more efficient methods. To address this, we propose to evaluate the SCA algorithm with the mutation and crossover mechanisms taken from the genetic algorithm (GA) for enhancement of the exploration capability.

3. Text Document Clustering

Text document clustering (TDC) has emerged as a rapidly growing research area in recent times [18]. TDC finds its application in numerous analytical tasks where a set of text documents needs to be grouped into classes or subsets of clusters [11]. Various techniques have been developed to ensure that the documents within a cluster exhibit high similarity while having minimal similarity with documents in other clusters [24]. The attribute values that represent the documents are used to assess their similarities and differences. TDC is a crucial issue in unsupervised learning as it deals with data partitioning in an unknown space, allowing for the organization of massive amounts of textual data and serving as a foundation for any subsequent unsupervised learning [11]. TDC approaches are employed to execute the classification of documents into corresponding categories or topics, without having samples from the related sets of documents in advance. This work focuses on partition clustering methods, where the clustering algorithm has the goal to perform the partitioning of a given dataset into smaller batches containing related clusters based on the minimizing or maximizing of the fitness function, irrespective of the ordered structure.

Text document clustering is considered an optimization problem, for which various algorithms for optimization provide different solutions. In population-based algorithms, each individual that belongs to the populace represent a potential solution for solving the clustering problem, where a vector in n dimensions determines the content of every document in the given dataset D, with each location corresponding to a document. The ith text is influenced by the ith placement of the solution. If the clusters' count is K, then each location in the individual has a value within the set (1,..., K), with each component corresponding to the one of the K centroids. The number of clusters typically predetermined.

The fitness value is calculated by evaluating each solution based on its positions. Each group of documents belongs to one of the K centroids C = (c1, c2,...., ck,...., cK), where ck is the centroid of the cluster k. The fitness function value for each possible solution is determined using the average similarity of documents to the cluster centroid (ASDC), as described in Equation (1). The similarity measure used in Equation (1) fitness can be adjusted to another similarity or distance measure.

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

The TDC is an NP-complete challenge of finding clusters in heterogenerous documents by reducing the fitness function (in this paper, min f(x) means reducing the Euclidean distance function). This part of the study covers the TDC problem definition and text document preparation.

3.1. Problem Definition

1 The TDC problem is formulated in the following way. Given a set of d documents, the objective is to

partition them into a predetermined number (K) of clusters, where Docs represents an array of documents $Docs = (d_1, d_2, ..., d_i, .., d_n)$. Each document di is associated with a unique number and collectively they form the set of documents in Docs. Each cluster has a corresponding cluster centroid (K_{cent}) represented as an array of terms with a weight factor f ($k_{cnt} = (k_{cnt1}, k_{cnt2}, ..., k_{cntj}, ..., k_{cntf})$), where k_{cnt} is the centroid of the k_{th} cluster k_{cnt1} indicates the number of occurrences of index 1 in the centroid of cluster k, and kcntf indicates the number of occurrences of all other centroid features(terms).

In order to determine a partition $k_{cnt} = (k_{cnt1}, k_{cnt2}, ..., k_{cntj}, ..., k_{cntf})$ it must satisfy these conditions:

1. $k_{cnt} \neq \varnothing$ every last one cluster can not be empty (i.e., each centroid has to captivate at least one document).
2. each $\cap\ k_{cnt'} = \varnothing$ if $K \neq K'$, $\bigcup_{k=1}^{K} k_{cnt} = 0$ each cluster has to consist of unique documents (i.e., Hard clustering).
3. The items that belong to a similar cluster have a high relation to each other, but the items that belong to different clusters are not like each other.

3.2. Data Preprocessing For Clustering Text

The following methods are the foundations for building a text clustering process.

3.2.1. Word Tokenization

Word tokenization is the division of an arrangement of characters to parts (words and phrases) that are called tokens, and reject certain characters such as punctuation. Tokens are stored in a list that is further processed Webster and Kit [45].

3.2.2. Word Filtering

Word filtering is done on documents when it is necessary to eliminate part of the words. The usual filtering is to discard stop words. Stop words are words that often show up in the text but are carriers of little or almost no information, and these are conjunctions, words, propositions, etc. Some examples are: 'in', 'as', 'are', 'about', 'yes' etc. Just as words that occur frequently and do not carry quality information, words that occur infrequently can be eliminated from the text because the information they carry is weak Reyaeiye et al. [37].

3.2.3. Lemmatization

Lemmatization is a method in which the morphological analysis of words given in different forms is considered, so words are formed in complex records, and the goal is to analyze them through the basic word. The method maps word forms into infinitives (verbs) and nominatives (nouns). Within the method, the POS procedure is applied, which classifies each word as a noun, pronoun, verb, or adjective.

3.2.4. Stemming

Stemming is a method that aims to extract the roots of a derived word. The algorithms that deal with this method depends on the language, and it is not possible to give a unique procedure given the syntactic-semantic complexity of each indiuage. Some examples are 'playing' and 'players' are treated as 'play'. 'Walking' is treated as 'walk'. An example of an algorithm that is one of the first algorithms of this type is given by Lovins [30] in his work on the development of stemming algorithms. The most used stemming algorithm is presented in Porter [34], however, it is for the English-speaking area.

3.2.5. Information Retrieval – IR

IR is the direction of textual data processing when resources (documents) are found from an assortment of disorganized data that meet the necessary information Sutton et al. [43]. This direction is focused mainly on facilitating access to information that analyzes information and searching for concealed patterns, that is the essential goal of data mining. However, creating quality access to information greatly facilitates understanding the results after the analysis as well as making decisions in which sense the textual data is analyzed.

3.2.6. TF-IDF

The term weighting, or the TF-IDF as traditionally alluded to in text clustering, is mainly used as a typical strategy to designate a weighting score to each document term Baeza-Yates and Ribeiro-Neto [7]. This strategy relies on the TF and IDF to speak for each document segment. TF-IDF system was mostly used to single out the document terms that are used as an objective function. The document batch is described by D as shown in Equation (2):

$$D = [d_1, d_2, \ldots, d_i, \ldots, d_n] \tag{2}$$

where n is the tally of documents in the documents batch D, and d_i is the *i-th* document, represented by Equation (3):

$$d = [w_{i,1}, w_{i,2}, \ldots, w_{i,t}] \quad (3)$$

3.2.7. Information Extraction – IE

IE is a text mining direction that aims to automatically extract information from unstructured or semi-structured text, as stated in Allahyari et al. [3]. It is often the origin of new text mining methods such as Extraction Entities and Name Entity Recognition.

3.2.8. Text Summarization – TS

TS is a text mining direction that summarizes text documents to obtain a concise overview of large documents and collections. Typically, the two groups of document summary methods are extractive summary when the summary contains information units isolated from the original text and abstract summary where the summary contains information that cannot appear in the original document, as described in Yao et al. [22].

3.2.9. Feature Extraction

Feature extraction is a method of building machine understandable attributes that have to be included in machine learning models. This is a type of dimensionality reduction where a large number of attributes from text mining are efficiently represented by a large variation of a data set. It is a vital foundation in text mining, used for transforming the original documents into a format that a data mining algorithm can proceed to use.

There are two basic approaches in features construction, and they are content-centric and non-content-centric approaches. Machine learning techniques evaluate the usefulness of sentences for summarization. Learning rules algorithms depend upon training sets so they can execute the learning with the aim that the training sets are formatted as a text section with binary annotations revealing if the sentence should be incorporated into the summary. This allows the use of binary classifiers where attributes are identified with distinct sentences and the label shows if it should take part in the summary.

More advanced approaches are considered operations with the words and sentences. It is possible to use a sentence breadth as an attribute as well as ratios between sentence breadths. The key idea is that summaries usually do not consist of short sentences. A set phrase feature was used that was assigned any number when the sentence had specific phrases. Next feature is a paragraph feature that indicates whether a sentence occurred in some position of the paragraph, such as the start, center, or conclusion of a paragraph. A particular feature was used, if an evaluation of a sentence based on the frequency of topics and words is larger than a specific threshold. Other features are considered types of words, such as nouns, verbs and adjectives. For each of them it is calculated a frequency. The approach that is powerful is the one that excerpts a lot of the features used for evaluating sentences (amongst other indicator features) and then proceeds to learn the gravity of a particular mix of features from the training data.

4. Suggested Clustering Approach

The clustering algorithm proposed in this study is introduced in this section. Initially, we describe the standard sine cosine algorithm (SCA as presented by Mirjalili [31]. Subsequently, we present an improved version of the SCA that addresses the limitations of the original algorithm. The standard SCA is affected by getting stuck in local optimums as it lacks sufficient search domain exploration. Hence, the proposed improved algorithm incorporates the mutation and crossover mechanisms for exploration from the Genetic Algorithm (GA) and it results in better exploration and exploitation trade-off.

4.1. Original Sine Cosine Algorithm

SCA has been introduced and developed by Mirjalili et al. in 2015. Initially, the solutions in the population are generated randomly. It is a population-based metaheuristics inspirited by the mathematical features of basic trigonometry. By utilizing the sine and cosine functions, the individuals are updated by the following equations:

$$Y_i^{t+1} = Y_i^t + r_1 \times sin(r_2) \cdot |r_3 \cdot P_i^t - Y_i^t| \quad (4)$$

$$Y_i^{t+1} = Y_i^t + r_1 \times cos(r_2) \cdot |r_3 \cdot P_i^t - Y_i^t|, \quad (5)$$

where the i-th element of the current individual is denoted by Y_i^t at iteration t, while the newly updated individual is referred by the Y_i^{t+1} notation. r_1, r_2 and r_3 are control parameters that are controlling the movement of the individuals. The destination point position is indicated by P_i^t.

A random pseudo-number (r_4) determines whether the equation based on sine or equation based on cosine function will be utilized according to the following criteria:

$$Y_i^{t+1} = \begin{cases} Y_i^{t+1} = Y_i^t + r_1 \cdot sin(r_2) \cdot \left| r_3 \cdot P_i^{*t} - Y_i^t \right|, & r_4 < 0.5 \\ Y_i^{t+1} = Y_i^t + r_1 \cdot cos(r_2) \cdot \left| r_3 \cdot P_i^{*t} - Y_i^t \right|, & r_4 \geq 0.5, \end{cases} \quad (6)$$

where the values of the random number $r4 \in [0,1]$.

The sine and cosine-based functions allow the exploitation around the current individuals. For balancing the exploration and exploitation, the control parameter r1 is updated according to the following formula:

$$r_1 = a - t\frac{a}{T}, \quad (7)$$

where the current iteration is denoted by t, T denotes the maximum number of rounds in a single run, while a is a constant.

The values of r2 are within the range $[0,2\pi]$, while r3 is a random number between -2 and 2.

4.2. Proposed Improved Sine Cosine Algorithm

The standard SCA is a simple and very efficient algorithm, however, it is poor at exploration and can easily stick in the local optima. To avoid the drawback, we incorporate the mutation and crossover operators from the genetic algorithm (GA), these methods allow better exploration of the problem space.

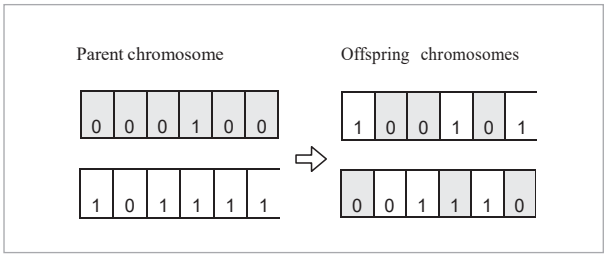
GA has been introduced as the algorithm inspired by the mechanism of nature's evolution. The new individuals in the algorithm are created by mutation and crossover operators. These two mechanisms are allowing the exploration of the problem space on a global scale.

In the proposed algorithm, the above-mentioned two mechanisms (mutation and crossover) are incorporated to help balancing the exploration and exploitation during the execution of the algorithm.

After the solutions are updated by the SCA updating equations Equation (6), they are evaluated with respect to their fitness, then sorted from the best to worst based on their fitness value. The entire population is separated in two portions, where the second part which contains the worst solutions are updated by the mutation and crossover operators.

From the second part of the population, the individuals are selected as parents to create new offspring by changing the parts of parents. The crossover operator is presented by an example in Figure 1. With respect to the new solutions, fitness is not evaluated, but the worst individuals are simply replaced by the solutions where genetic operators were applied. The reason is the following – the new solution maybe has lower fitness that the previous one, but it has found the optimal region of the search space (not the local optimum), which is the main goal of the exploration. The uniform crossover is applied, on the level of each gene (parameter of the objective function) with probability p_c.

Figure 1
Crossover operator



After the crossover operation, the mutation operator is utilized that occurs based on the mutation probability. For each offspring the mutation on each gene is performed with probability p_m. You may refer to Holland et al. [20] for more details about the GA. The GA or specific operations in the genetic algorithm are used successfully in different applications Cuk et al. [17]. The probabilities p_c and p_m were determined empirically, as according no free luch theorem (NFL) Wolpert and Macready [46] a universal solution does not exist, hence it must be determined for each problem individually, and the values are provided in the experimental section.

For metaheuristic algorithms, complexity is typically regarded in terms of fitness function evaluations

(FFEs) Chopard et al. [16], as it is observed as the most intensive operation. The proposed algorithm is not introducing any additional complexity to the elementary SCA metaheuristics, as during the replacement of the worst solutions with the offspring solutions fitness is not evaluated and compared, but calculated normally at the end of the each iteration. Therefore, the complexity of the SCA-GA algorithm is the same as SCA, stated by O(N)=N+N·T, where N s the count of individuals in the population, while T denotes the count of iterations.

The proposed improved algorithm is named as SCA-GA and its pseudocode is presented in Algorithm 1.

Initialize the individuals randomly;

Evaluate all individuals based on the fitness value; Save the best solution P;

while t < T **do**

Update the values of the control parameters r_1, r_1, r_3 and r_4; Update the individuals by Equation (6);

Evaluate and sort the solutions according to their fitness value; Apply crossover and mutation on the worst solutions;

Evaluate and sort the solutions according to their fitness value;

end

Return the best solution obtained so far as the global optimum;

Algorithm 1: Pseudocode of the proposed SCA-GA method

5. Simulation Results and Discussion

In this section, the simulation setup and results are presented. First, the newly proposed method is tested on the set of unconstrained benchmark functions, specifically on CEC2017 Wu et al. [47] test-suite. Subsequently, the proposed approach is applied for text document clustering.

5.1. CEC 2017 Simulations

The CEC benchmark functions are extensively utilized test problems by scientific researchers for evaluating the performance and proving the effectiveness of their algorithms. In the present work, the proposed SCA-GA algorithm is evaluated on CEC 2017 problem sets. The CEC 2017 test suite includes 30 benchmark functions with different characteristics. The first three functions (F1-F3) are unimodal functions, F4-F10 simple multimodal functions, F11-F20 hybrid functions, and the last 10 functions (F21-F30) are composition functions. The search range for each function is between -1 and 1. The definition and properties of each function are detailed in Wu et al. [47].

Table 1 presents the obtained results, in terms of mean and standard deviation, of the original SCA and the newly proposed SCA-GA, as well as the comparison with other metaheuristic approaches, including Haris hawks' optimization (HHO), improved HHO (IHHO), differential evolution (DE), grasshopper optimization algorithm (GOA), gray wolf optimizer (GWO), moth-flame optimizer (MFO), multi-verse optimizer (MVO), particle swarm optimization (PSO) and whale optimization algorithm (WOA). The results of the comparable approaches are taken from Hussien and Amin [21] where the experiments are conducted under similar setup and conditions. Because of the unstable behavior Gupta and Deep [19], the second function (F2) was excluded from the simulations, consequently, the results are not reported in Table 1.

In the single objective optimization experiments, the population size is set up to 30, the maximum number of fitness function evaluations (FFE) is 15,000, and the simulations are executed in 50 independent runs. The parameters for SCA-GA were set as follows: p_c = 0.1 and p_m=0.05, and these values were determined empirically.

The best-obtained results are indicated by boldface in Table 1, additionally, if two or more algorithms obtained the same best results, the value is underlined.

Based on the obtained results, we can conclude that the proposed SCA-GA method is very competitive, it achieved the best result on 26 functions out of the 29 benchmark functions. To better investigate the significance of the proposed method over other state-of-the-art methods and show the significant differences, non-parametric statistical tests are applied, specifically, Friedman and Friedman aligned rank tests are utilized. In the non-parametric statistical tests, a rank is assigned to each algorithm, where rank 1 indicates the best performing algorithm, on the other hand, 11 indicates the worst-performing algorithm. Friedman test is a pairwise comparison procedure, while the Friedman aligned rank test is a multiple comparison

Table 1
Statistical results of the proposed method and other comparative methods on CEC 2017 test-suit

Algorithm	F1		F2		F3		F4		F5	
	mean	STD	mean	STD	mean	STD	mean	STD	mean	STD
IHHO	1.86E+2	26.921	N/A	N/A	3.02E+2	52.152	**4.03E+2**	2.607	5.05E+2	**3.251**
HHO	1.75E+6	4.29E+5	N/A	N/A	6.71E+2	3.24E+2	4.37E+2	53.631	5.35E+2	24.927
DE	7.54E+7	1.71E+7	N/A	N/A	4.59E+3	1.35E+3	4.29E+2	8.530	5.52E+2	6.232
GOA	1.56E+5	5.24E+4	N/A	N/A	3.05E+2	61.300	4.15E+2	19.48	5.25E+2	16.803
GWO	1.53E+7	4.85E+6	N/A	N/A	3.57E+3	2.77E+3	4.09E+2	10.705	5.19E+2	8.543
MFO	7.17E+6	2.18E+7	N/A	N/A	9.04E+3	9.31E+3	4.20E+2	27.727	5.31E+2	12.860
MVO	1.79E+4	7.99E+3	N/A	N/A	3.05E+2	46.451	4.06E+2	**1.392**	5.17E+2	9.888
PSO	9.49E+4	8.42E+2	N/A	N/A	3.49E+2	65.409	4.07E+2	10.318	5.26E+2	7.305
WOA	4.27E+7	3.81E+6	N/A	N/A	5.16E+3	4.22E+2	4.61E+2	69.033	5.51E+2	17.46
SCA	1.15E+8	5.91E+7	N/A	N/A	4.03E+3	8.42E+2	4.85E+2	47.271	5.59E+2	9.352
SCA-GA	**1.53E+2**	**22.381**	N/A	N/A	**3.01E+2**	38.192	**4.03E+2**	3.865	**5.04E+2**	3.563

Algorithm	F6		F7		F8		F9		F10	
	mean	STD	mean	STD	mean	STD	mean	STD	mean	STD
IHHO	**6.00E+2**	**0.082**	7.49E+2	10.041	8.11E+2	6.526	1.13E+3	85.42	1.69E+3	1.31E+2
HHO	6.38E+2	12.320	7.96E+2	18.921	8.29E+2	5.700	1.44E+3	1.24E+2	2.03E+3	3.42E+2
DE	6.28E+2	4.744	8.01E+2	10.373	8.62E+2	6.873	1.76E+3	1.48E+2	2.09E+3	2.01E+2
GOA	6.08E+2	10.295	7.32E+2	11.375	8.31E+2	14.512	9.97E+2	93.212	1.96E+3	3.17E+2
GWO	6.01E+2	1.909	7.35E+2	16.343	8.16E+2	**5.053**	9.14E+2	12.11	1.76E+3	3.10E+2
MFO	6.02E+2	2.411	7.46E+2	22.655	8.29E+2	13.786	1.23E+3	2.76E+2	2.02E+3	3.27E+2
MVO	6.03E+2	4.365	7.30E+2	11.278	8.25E+2	12.216	**9.00E+2**	0.012	1.82E+3	3.60E+2
PSO	6.10E+2	3.539	7.26E+2	**9.008**	8.19E+2	5.982	**9.00E+2**	**0.003**	**1.50E+3**	2.84E+2
WOA	6.36E+2	13.695	7.82E+2	23.692	8.45E+2	17.470	1.54E+3	3.94E+2	2.19E+3	3.16E+2
SCA	6.24E+2	4.105	7.84E+2	13.299	8.47E+2	7.577	1.03E+3	85.98	2.51E+3	2.18E+2
SCA-GA	**6.00E+2**	0.093	**7.24E+2**	10.839	**8.09E+2**	6.385	**9.77E+2**	15.852	**1.49E+3**	**1.18E+2**

Algorithm	F11		F12		F13		F14		F15	
	mean	STD	mean	STD	mean	STD	mean	STD	mean	STD
IHHO	1.12E+3	13.523	4.25E+5	3.05E+5	4.42E+3	2.18E+3	**1.42E+3**	**1.651**	2.15E+3	5.65E+2
HHO	1.16E+3	45.729	2.56E+6	1.13E+6	1.92E+4	1.16E+4	1.83E+3	2.41E+2	8.63E+3	5.55E+2
DE	1.14E+3	36.317	9.15E+4	6.58E+4	**1.35E+3**	78.355	1.46E+3	11.826	**1.51E+3**	18.454
GOA	1.17E+3	58.009	2.24E+6	1.15E+6	1.65E+4	1.13E+4	2.93E+3	1.15E+3	6.48E+3	4.32E+3
GWO	1.34E+3	183.524	1.31E+6	1.54E+6	1.26E+4	7.82E+3	3.19E+3	1.82E+3	5.63E+3	3.16E+3
MFO	1.23E+3	107.133	2.23E+6	4.81E+6	1.61E+4	1.39E+4	8.42E+3	5.42E+3	1.25E+4	1.02E+4
MVO	1.14E+3	27.331	1.52E+6	1.41E+6	9.89E+3	2.55E+3	2.15E+3	1.03E+3	4.05E+3	2.45E+3
PSO	**1.10E+3**	3.727	4.35E+4	1.26E+4	1.01E+4	7.23E+3	1.49E+3	88.291	1.81E+3	3.75E+2
WOA	1.22E+3	82.415	4.85E+6	5.12E+6	1.57E+4	1.38E+4	3.42E+3	9.82E+2	1.42E+4	9.88E+3
SCA	1.24E+3	96.535	2.41E+7	2.05E+7	6.43E+4	4.69E+4	1.99E+3	4.31E+2	3.21E+3	1.41E+3
SCA-GA	**1.10E+3**	**2.825**	**3.56E+4**	**1.15E+4**	1.82E+3	**50.421**	1.97E+3	19.382	**1.50E+3**	**12.854**

Algorithm	F11		F12		F13		F14		F15	
	mean	STD	mean	STD	mean	STD	mean	STD	mean	STD
IHHO	1.73E+3	59.44	1.73E+3	**7.519**	4.79E+3	1.68E+3	**1.90E+3**	**6.993**	2.02E+3	19.561
HHO	1.89E+3	1.47E+2	1.79E+3	65.751	2.02E+4	1.41E+4	1.71E+4	1.21E+4	2.23E+3	86.017
DE	1.69E+3	**41.15**	1.77E+3	19.514	**1.84E+3**	23.298	2.75E+3	8.35E+2	2.05E+3	23.711
GOA	1.78E+3	1.76E+2	1.83E+3	1.21E+2	1.63E+4	1.31E+4	3.25E+3	1.95E+3	2.15E+3	74.824
GWO	1.79E+3	1.11E+2	1.77E+3	38.759	2.55E+4	1.84E+4	2.75E+4	2.38E+4	2.09E+3	73.994
MFO	1.85E+3	15.23E+2	1.78E+3	65.311	2.21+4	1.39E+4	7.81E+3	6.15E+3	2.13E+3	72.321
MVO	1.80E+3	1.44E+2	1.80E+3	46.126	2.03E+4	1.25E+4	4.63E+3	2.62E+3	2.12E+3	86.303
PSO	**1.65E+3**	65.364	1.72E+3	16.123	7.63E+3	4.46E+3	3.13E+3	2.05E+3	2.06E+3	35.410
WOA	1.96E+3	14.92E+2	1.82E+3	73.459	2.13E+4	1.95E+2	2.07E+5	1.16E+5	2.19E+3	1.11E+2
SCA	1.73E+3	95.425	1.80E+3	25.303E	8.77E+4	9.23E+2	1.15E+4	1.44E+3	2.14E+3	46.855
SCA-GA	**1.69E+3**	75.391	**1.70E+3**	11.698	**1.84E+3**	19.482	**1.90E+3**	9.365	**2.00E+3**	10.483

Algorithm	F11		F12		F13		F14		F15	
	mean	STD	mean	STD	mean	STD	mean	STD	mean	STD
IHHO	2.20E+3	**4.615**	2.28E+3	17.820	2.59E+3	14.213	2.68E+3	1.31E+2	2.87E+3	85.338
HHO	2.35E+3	53.711	2.32E+3	25.234	2.69E+3	35.522	2.82E+3	93.623	2.95E+3	49.573
DE	2.25E+3	78.104	2.29E+3	17.513	2.63E+3	15.163	**2.66E+3**	69.502	2.91E+3	**15.543**
GOA	2.30E+3	56.877	2.38E+3	1.08E+2	2.64E+3	23.536	2.73E+3	57.833	2.93E+3	32.598
GWO	2.30E+3	32.884	2.31E+3	57.573	2.62E+3	13.862	2.74E+3	25.132	2.94E+3	28.256
MFO	2.32E+3	29.255	2.35E+3	93.557	2.63E+3	11.327	2.75E+3	76.435	2.96E+3	37.776
MVO	2.32E+3	11.839	2.33E+3	1.11E+2	2.65E+3	**10.445**	2.74E+3	18.246	2.92E+3	84.256
PSO	2.27E+3	49.783	2.33E+3	1.03E+2	2.60E+3	72.300	2.70E+3	76.143	2.90E+3	33.735
WOA	2.34E+3	60.021	2.48E+3	2.45E+2	2.66E+3	29.838	2.77E+3	85.902	2.98E+3	1.03E+2
SCA	2.29E+3	65.229	2.41E+3	66.636	2.67E+3	45.449	2.78E+3	**11.548**	2.98E+3	37.291
SCA-GA	**2.15E+3**	29.545	**2.24E+3**	**15.385**	**2.49E+3**	12.341	**2.59E+3**	85.3921	**2.79E+3**	59.294

Algorithm	F26		F27		F28		F29		F30	
	mean	STD	mean	STD	mean	STD	mean	STD	mean	STD
IHHO	2.93E+3	1.66E+2	3.19E+3	33.657	3.30E+3	48.694	**3.20E+3**	28.982	2.30E+4	1.45E+4
HHO	3.62E+3	5.39E+2	3.18E+3	51.306	3.41E+3	1.02E+2	3.39E+3	85.653	1.43E+6	1.31E+6
DE	2.95E+3	**95.929**	**3.07E+3**	**2.558**	3.28E+3	**27.035**	**3.21E+3**	35.216	3.65E+5	2.31E+5
GOA	3.01E+3	3.65E+2	3.11E+3	25.326	3.31E+3	1.53E+2	3.27E+3	75.411	5.29E+5	3.89E+5
GWO	3.36E+3	5.05E+2	3.10E+3	13.541	3.42E+3	1.33E+2	3.22E+3	49.822	6.17E+5	4.88E+5
MFO	3.05E+3	1.13E+2	3.09E+3	5.722	3.21E+3	93.459	3.26E+3	55.593	6.36E+5	5.93E+5
MVO	3.15E+3	2.77E+2	3.10E+3	21.875	3.36E+3	1.23E+2	3.26E+3	75.139	4.62E+5	4.07E+5
PSO	2.95E+3	2.55E+2	3.12E+3	31.830	3.32E+3	1.35E+2	3.21E+3	62.374	1.13E+6	1.09E+6
WOA	3.37E+3	2.92E+2	3.17E+3	48.124	3.46E+3	1.65E+2	3.46E+3	1.21E+2	1.29E+6	7.53E+5
SCA	3.15E+3	1.82E+2	3.13E+3	13.152	3.38E+3	89.25	3.25E+3	48.339	1.49E+6	9.77E+5
SCA-GA	**2.77E+3**	3.61E+2	3.12E+3	85.361	**3.19E+3**	51.32	**3.20E+3**	19.831	**3.59E+3**	**5.31E+3**

procedure. The result of the two non-parametric statistical procedures is reported in Table 2.

The results in Table 2 proofs the efficiency and effectiveness of the proposed SCA-GA, where it achieved rank 1 on the Friedman test, as well as Friedman aligned rank tests.

Table 2
Friedman and Friedman aligned rank tests results

Algorithm	Friedman Test Average	Friedman TestRank	Friedman Aligned Ranking Average	Friedman Aligned Ranking Rank
IHHO	3.05	2	94.36	2
HHO	8.64	9	202.62	9
DE	5.05	4	140.50	5
GOA	6.45	7	159.93	6
GWO	6.14	6	169.76	7
MFO	7.24	8	187.95	8
MVO	5.48	5	130.78	4
PSO	3.90	3	109.19	3
WOA	9.67	11	266.55	11
SCA	8.76	10	218.81	10
SCA-GA	1.62	1	79.55	1
Friedman Statistic	166.82		112.91	
p-value	1.11E-16		0.00	

5.2. Text Document Clustering Experiment

After evaluating and proofing the efficiency of the proposed method on unconstrained benchmark functions, the SCA-GA is applied for text document clustering task. The text document clustering experiment is conducted on 6 standard benchmark datasets:

- The Centre for Speech Technology Research (CSTR)
- The 20Newsgroups dataset (20Newsgroups)
- Tr41
- Tr12
- Wap
- Classic4

The CSTR dataset is formed in 1984, it contains 299 documents from technical report abstracts of four different research areas, which are robotics, systems, artificial intelligence, and theory. The 20Newsgroups dataset consists of approximately 20,000 articles partitioned into 20 clusters. In this research, three different topics are used from 20Newsgroups, which are talking.politics.misc, comp.windows.x, and rec.autos. The Tr41 dataset is partitioned into 10 clusters, and it contains 878 documents. The Tr12 dataset consists of 313 documents and it has 8 classes. The Wap dataset consists of 1,560 datasets belonging to 20 distinct clusters. The sixth dataset, Classic4, has 2,000 the documents and four classes, each class has 500 documents.

The following metrics are used to evaluate the model: accuracy (Equation (8)), precision (Equation (9)), recall (Equation (10)), F-measure (Equation (11)), purity (Equation (12)), and entropy.

The clusters are groups of documents, where the documents that belong to the same cluster are similar to each other with respect to content. Classes refer to the classes of the input documents, which may or may not be assigned to the correct number. The correctly assigned clusters in a given dataset are measured by the accuracy and it is calculated as follows:

$$AC = \frac{1}{n}\sum_{j=1}^{k} n_{i,j}, \tag{8}$$

where n denotes the total number of documents, while the correctly assigned documents are in cluster j of class i.

The correct classes in the cluster over all classes are calculated by precision as follows:

$$P(i,j) = \frac{n_{i,j}}{n_i}, \tag{9}$$

where i denotes a class, the cluster is referred by j, the total number of documents in j-th cluster is indicated by n_j. The correctly assigned class i in the j cluster is denoted by $n_{i,j}$.

The ratio of correct document assignment over the total number of documents in the given class is measured by the recall:

$$R(i,j) = \frac{n_{i,j}}{n_i}, \tag{10}$$

where n_i indicates to the total number of documents in class i. $n_{i,j}$ denotes the correctly assigned class i in the j cluster.

The Harmonic mean of the precision and recall are expressed by the F-measure, according to the following formula:

$$F(i,j) = \frac{2\times P(i,j)\times R(i,j)}{P(i,j)+R(i,j)}, \quad (11)$$

where P(i,j), and R(i,j) denotes the precision, and recall, respectively.

Purity calculates the percentage of each cluster, where the documents are assigned from the correct class to a cluster by using the following equation:

$$purity = \frac{1}{n}\sum_{i=1}^{k} m\,ax(i,j), \quad (12)$$

where i denotes the class, j denotes the cluster and n denotes the total number of documents. The best value of purity is 1, while the worst is 0.

The distribution of documents of class labels in each cluster is measured by the entropy, and it is calculated as follows:

$$E(j) = -\sum_{i=1} p\,(i,j)\log p(i,j), \quad (13)$$

where the probability of class i in cluster j of a document is denoted by p(i,j), and E(j) denotes the entropy of cluster j.

If the entropy value is closer to 0, it indicates a better result. The entropy of all clusters is calculated by the following formula:

$$E = \sum_{j=1}^{k} \frac{n_j}{n} E(j). \quad (14)$$

The K-means procedure is included in the proposed method for TDC problem simulations. In the algorithm, initially, the individuals in the populations are produced in a random manner within the provided lower and upper bounds. The individuals are encoded as follows:

$$X_i = (C_1, C_2, \ldots, C_i, \ldots, C_k) \quad (15)$$

in the population where the cluster centroid is provided by C_i, while k stands for the cluster number.

According to the calculated distance from the centroid to the documents, the documents are being assigned to the closest centroid, and it is calculated as follows:

$$dist(d_j, c_k) = \sqrt{\sum_{i=1}^{t} \left|d_{ij} - c_{ik}\right|^2}, \quad (16)$$

where the number of terms are denoted by t, the documents are denoted by d, the centroid is indicated by c.

The fitness of a solution is determined by the average distance between a document and a centroid, and it is calculated by the following equation:

$$f(x) = \frac{\sum_{i=1}^{k}\left(\frac{1}{n_i}\sum_{j}^{n_i} dist(d_{ij}, c_i)\right)}{K}. \quad (17)$$

In the given notation K represents the total count of clusters, n_i denotes the count of documents in the i-th cluster, $d_{i,j}$ represents the j-th document in cluster i, and $c_{i,j}$ is the centroid of the i-th cluster. The function *dist()* computes the distance between the document $d_{i,j}$ and the centroid c_i of the corresponding cluster.

For the TDC experiment, the number of individuals in the populace was set to twenty, and the maximum number of rounds was equal to 1000. The experiment is performed in 30 separate runs.

The proposed method and the standard SCA algorithm are compared with various state-of-the-art methods, such as K-mean, K-mean++, DBSCAN, Agglomerative, Spectral, KHA, HS, PSO, GA, MVO, H-PSO, H-GA, H-MVO1, H-MVO2, FFO, and HEFF. The results of the comparable methods are obtained from Abasi et al. [1] and Bezdan et al. [9], respectively.

The results and comparison are shown in Tables 3-8.

Based on the obtained results from the simulation and comparative analysis, we can conclude that the proposed SCA-GA approach is very robust and competitive in text document clustering. On five datasets SCA-FA obtained the best accuracy, and only in the case of the tr41 dataset SCA is the best performing method, however, SCA-GA's accuracy is very close. Based on the statistical measures. The second-best approach is the SCA.

In order to provide a comprehensive overview of the experimental outcomes presented in Tables 3-8, we also present a comparison of metrics across all methods and datasets in Figures 2-3.

Table 3
Results of the **CSTR dataset**

Method	Accuracy	Precision	Recall	F-measure	Purity	Entropy
K-mean	0.3573	0.4091	0.3092	0.346	0.3525	0.8201
K-mean++	0.4355	0.3953	0.4076	0.3546	0.4096	0.5246
DBSCAN	0.4005	0.3393	0.4256	0.3046	0.4076	0.4586
Agglomerative	0.436	0.4423	0.4666	0.3266	0.4816	0.5076
Spectral	0.4319	0.3597	0.4925	0.3971	0.4485	0.4893
KHA	0.3649	0.4213	0.5355	0.4139	0.3874	0.4344
HS	0.4464	0.4235	0.506	0.3377	0.4355	0.4786
PSO	0.4356	0.534	0.436	0.4819	0.4953	0.6199
GA	0.3399	0.4417	0.3418	0.3886	0.4050	0.717
MVO	0.4593	0.5715	0.4829	0.5244	0.5685	0.5207
H-PSO	0.5494	0.6065	0.56	0.5577	0.6135	0.5076
H-GA	0.5056	0.613	0.495	0.5859	0.5743	0.5104
H-MVO1	0.5683	0.6395	0.5079	0.5774	0.5015	0.4577
H-MVO2	0.5779	0.6497	0.5568	0.5936	0.5980	0.4010
FFO	0.5859	0.5757	**0.5944**	0.5849	0.5860	0.3397
HEFFF	0.5964	0.5873	0.5641	0.5755	0.6140	0.3361
SCA	0.5972	0.5892	0.5799	0.5896	0.6267	0.3295
SCA-GA	**0.5988**	**0.5985**	0.5832	**0.5991**	**0.6395**	**0.3122**

Table 4
Results of the 20Newsgroups dataset

Method	Accuracy	Precision	Recall	F-measure	Purity	Entropy
K-mean	0.318	0.3121	0.31	0.3406	0.3741	0.8028
K-mean++	0.3784	0.3652	0.3662	0.3619	0.4134	0.6611
DBSCAN	0.3038	0.3094	0.3017	0.3193	0.3027	0.7473
Agglomerative	0.4055	0.399	0.3576	0.3548	0.4417	0.5953
Spectral	0.3633	0.3424	0.328	0.3136	0.311	0.6125
KHA	0.3216	0.3829	0.3136	0.2996	0.3421	0.6767
HS	0.3122	0.3601	0.317	0.3214	0.3355	0.6481
PSO	0.3498	0.4134	0.3497	0.3803	0.4097	0.7723
GA	0.3676	0.4209	0.3676	0.3936	0.4081	0.7547
MVO	0.4044	0.4392	0.3842	0.4109	0.4344	0.7121
H-PSO	0.4442	0.4821	0.422	0.4454	0.4725	0.6631
H-GA	0.4308	0.4814	0.4887	0.4433	0.4837	0.6833
H-MVO1	0.5174	0.5102	0.4332	0.5169	**0.5104**	0.5811
H-MVO2	0.5326	0.5619	0.4876	0.5376	*0.4591*	*0.5407*
FFO	0.5500	0.5376	0.5500	0.5437	*0.5032*	*0.3299*
HEFFF	0.5833	0.5668	0.5833	0.5750	*0.5094*	*0.3267*
SCA	0.5848	0.5697	0.5812	**0.5895**	*0.5091*	*0.3587*
SCA-GA	**0.5996**	**0.5821**	**0.5961**	0.5620	0.5098	**0.3118**

Table 5
Results of the tr12 dataset

Method	Accuracy	Precision	Recall	F-measure	Purity	Entropy
K-mean	0.2971	0.3522	0.2944	0.3222	0.3908	0.7138
K-mean++	0.3795	0.4215	0.3778	0.4176	0.4808	0.5094
DBSCAN	0.3045	0.3218	0.3234	0.4048	0.4728	0.501
Agglomerative	0.4481	0.4923	0.4341	0.409	0.5553	0.3978
Spectral	0.3373	0.4508	0.3055	0.3781	0.4132	0.4932
KHA	0.3357	0.3748	0.3099	0.2916	0.3803	0.5183
HS	0.3776	0.4385	0.3453	0.447	0.4986	0.4517
PSO	0.4075	0.4298	0.4264	0.4278	0.4878	0.572
GA	0.3677	0.4128	0.3549	0.3826	0.4513	0.6233
MVO	0.4485	0.5075	0.4398	0.4706	0.5448	0.5224
H-PSO	0.4796	0.5355	0.3893	0.586	0.5876	0.5757
H-GA	0.5205	0.4798	0.4974	0.5428	**0.6038**	0.638
H-MVO1	0.5465	0.6025	0.5138	0.5776	0.5948	0.4224
H-MVO2	0.5617	0.6398	0.5109	0.5956	0.6063	**0.3753**
FFO	0.5769	0.5992	0.5771	0.5879	0.535	0.6663
HEFFF	0.6398	0.6295	**0.6284**	0.6290	0.5744	0.6965
SCA	0.6421	0.6371	0.6223	0.6351	0.5635	0.3795
SCA-GA	**0.6497**	**0.6452**	0.6234	**0.6392**	0.5471	**0.3549**

Table 6
Results of the tr41 dataset

Method	Accuracy	Precision	Recall	F-measure	Purity	Entropy
K-mean	0.4126	0.3945	0.3813	0.3876	0.4108	0.5874
K-mean++	0.426	0.3769	0.3559	0.4299	0.5471	0.4745
DBSCAN	0.3704	0.4242	0.4291	0.4152	0.6025	0.4776
Agglomerative	0.461	0.3445	0.3487	0.4187	0.476	0.4821
Spectral	0.3787	0.363	0.348	0.3691	0.4866	0.5386
KHA	0.4054	0.4191	0.4579	0.4137	0.6063	0.4041
HS	0.359	0.371	0.3888	0.3701	0.4863	0.4539
PSO	0.487	0.4505	0.4497	0.4497	0.579	0.5391
GA	0.432	0.414	0.4008	0.4071	0.5603	0.5469
MVO	0.463	0.4569	0.4419	0.4569	0.6081	0.5355
H-PSO	0.474	0.501	0.4768	0.4701	0.5273	0.5029
H-GA	0.594	0.5245	0.5127	0.5647	0.642	0.5371
H-MVO1	0.568	0.5489	0.5079	0.5119	0.6421	0.3965
H-MVO2	0.6150	0.569	0.5178	0.5201	**0.6673**	0.3669
FFO	0.6170	0.6111	0.6132	0.6122	0.5475	0.3160
HEFFF	0.6487	0.6341	0.6420	**0.6380**	0.5791	0.3130
SCA	**0.6541**	0.with6485	0.6792	0.6281	0.5932	0.3028
SCA-GA	0.6538	**0.6721**	**0.6918**	0.6319	0.6715	**0.3005**

Table 7
Results of the Wap dataset

Method	Accuracy	Precision	Recall	F-measure	Purity	Entropy
K-mean	0.5012	0.4626	0.4011	0.4315	0.4759	0.7044
K-mean++	0.4937	0.4913	0.3508	0.4462	0.5797	0.5961
DBSCAN	0.4846	0.5004	0.4176	0.4878	0.4007	0.5706
Agglomerative	0.4933	0.4479	0.4471	0.4487	0.5445	0.4875
Spectral	0.4905	0.4454	0.3423	0.4141	0.5142	0.5468
KHA	0.5191	0.4773	0.4106	0.4131	0.5939	0.5595
HS	0.5032	0.463	0.4447	0.4611	0.5589	0.547
PSO	0.5623	0.5249	0.4811	0.5017	0.6125	0.5765
GA	0.5316	0.5314	0.4706	0.4998	0.4917	0.6216
MVO	0.5291	0.5213	0.4496	0.4831	0.6069	0.6625
H-PSO	0.5532	0.545	0.4907	0.5521	0.6939	0.631
H-GA	0.6523	0.5749	0.5671	0.6187	**0.7315**	0.6585
H-MVO1	0.5931	0.6063	0.5906	0.5911	0.6849	0.4365
H-MVO2	0.6176	0.6344	0.5855	0.6105	0.7107	0.4196
FFO	0.5769	0.5992	0.5771	0.5879	0.535	0.6663
HEFFF	0.6398	0.6295	0.6284	0.6290	0.5985	0.6965
SCA	0.6487	0.6491	0.63954	0.6397	0.6587	0.4598
SCA-GA	**0.6598**	**0.6577**	**0.6585**	**0.6485**	0.6985	**0.4029**

Table 8
Results of the classic4 dataset

Method	Accuracy	Precision	Recall	F-measure	Purity	Entropy
K-mean	0.5858	0.5699	0.5259	0.5472	0.5938	0.5601
K-mean++	0.6799	0.688	0.7028	0.692	0.6227	0.354
DBSCAN	0.6206	0.5708	0.5796	0.6173	0.6089	0.4854
Agglomerative	0.6683	0.6449	0.6224	0.6281	0.5812	0.4033
Spectral	0.6326	0.662	0.6528	0.6764	0.6352	0.4002
KHA	0.5761	0.6246	0.555	0.5529	0.555	0.5321
HS	0.5963	0.6064	0.5674	0.5787	0.5983	0.4946
PSO	0.6363	0.6604	0.6164	0.6377	0.6243	0.5306
GA	0.6621	0.6726	0.632	0.6519	0.632	0.5781
MVO	0.7043	0.6919	0.6844	0.6881	0.6742	0.5113
H-PSO	0.6853	0.7354	0.6754	0.6297	0.6413	0.5656
H-GA	0.7273	0.7774	0.6944	0.6887	0.6733	0.6166
H-MVO1	0.7163	0.7129	0.6844	0.6951	0.7042	0.3553
H-MVO2	0.7271	0.7226	0.705	0.7079	0.7320	0.3341
FFO	0.7296	0.6102	0.7236	0.6621	0.5501	0.262
HEFFF	0.7459	0.6622	0.7467	0.7029	0.6311	0.288
SCA	0.7598	0.7632	0.7598	0.7125	0.6987	0.2681
SCA-GA	**0.6598**	**0.7895**	**0.7987**	**0.7532**	**0.7525**	**0.2789**

Figure 2
Comparison of accuracy, precision, and recall over all comparable methods

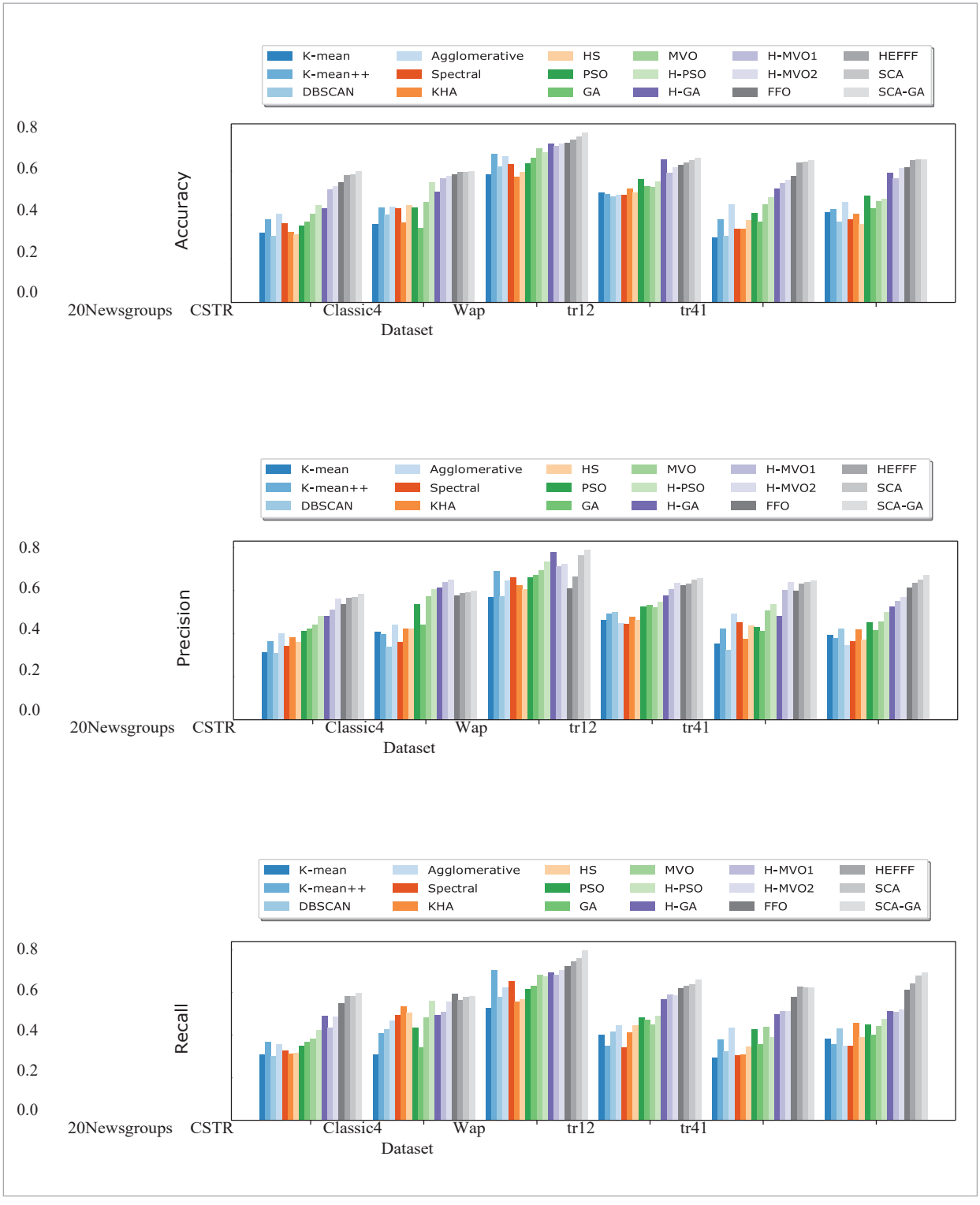
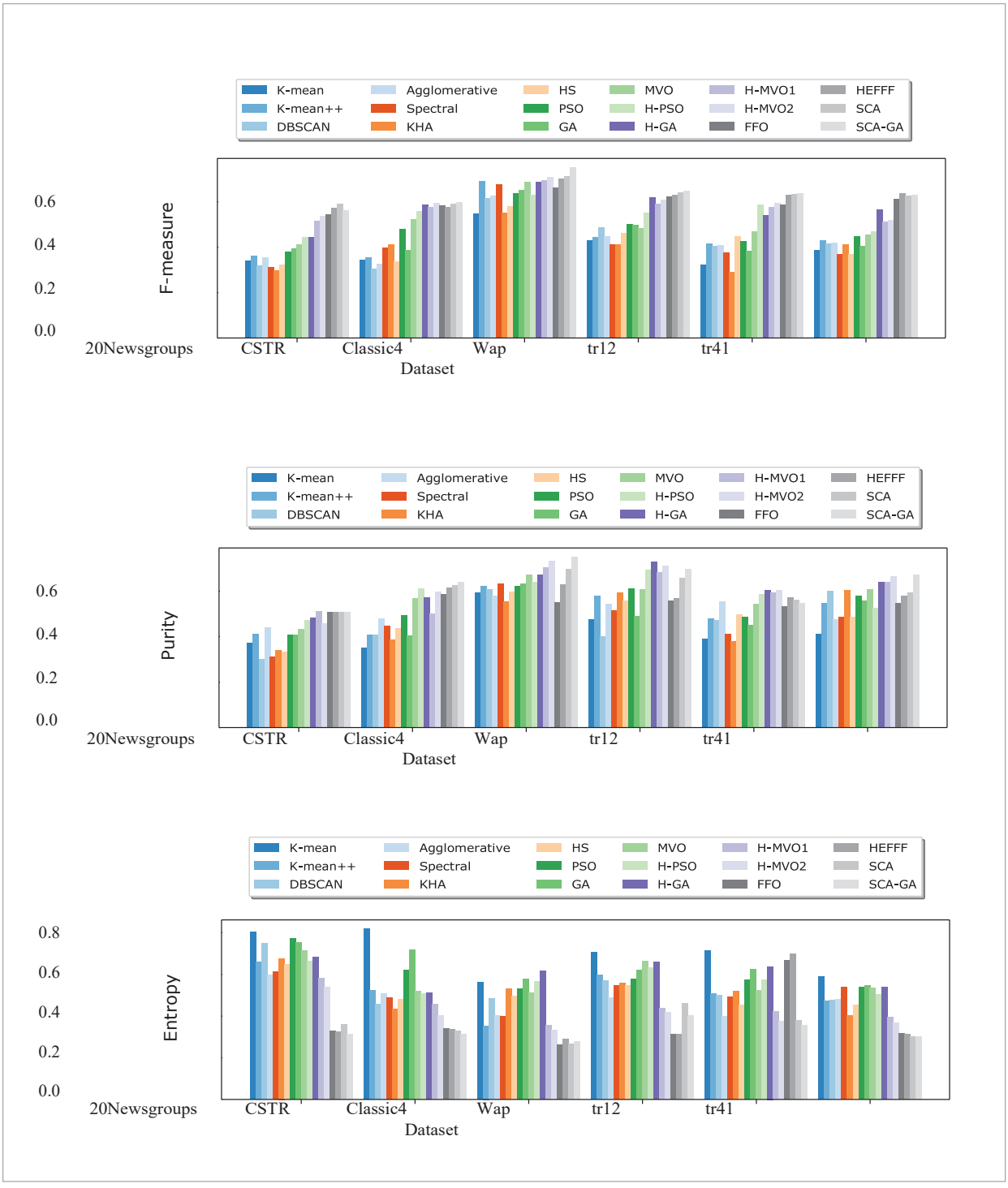


Figure 3

Comparison of F-measure, purity and entropy over all comparable methods

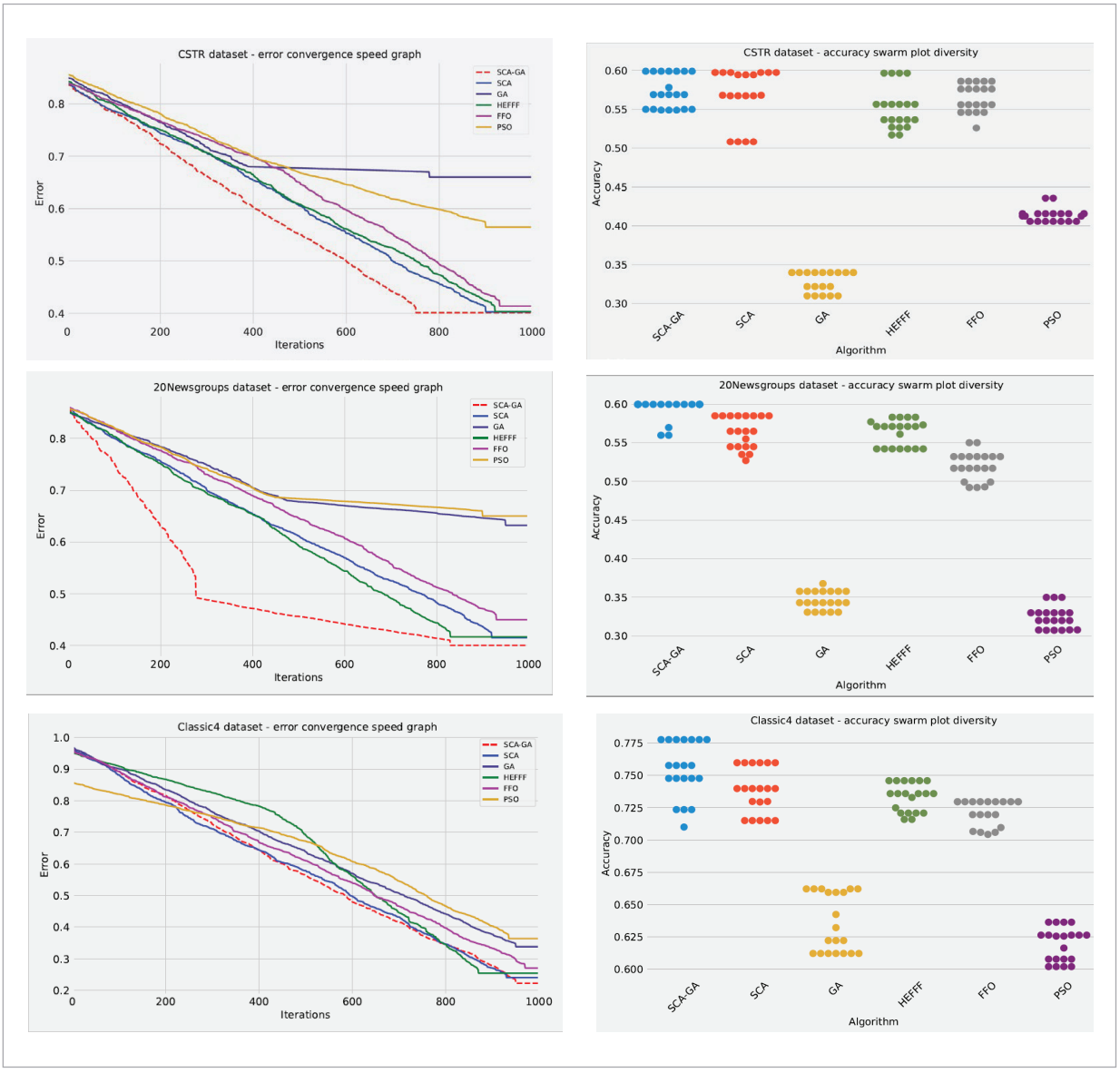


Finally, in order to better visualize achieved results, error convergence speed graph over 1000 iterations and swarm plot diagram for final population diversity in terms of accuracy in the final iteration of the best run for CSTR, 20Newsgroups and Classic4 datasets for proposed SCA-GA, basic SCA and GA, as well as for few other approaches are depicted in Figure 4.

From the provided figure, it can be clearly seen that the proposed SCA-GA exibits the best convergence speed among all compared methods. Additionally, the SCA-Ga also proved as the most efficient optimizer since almost all solutions from population have converged around the region of the best individual.

Figure 4

Error convergence speed graphs and population diversity of some methods for CSTR, 20Newsgroups and Classic4 experiments



6. Conclusions

In this manuscript, a novel improved SCA algorithm suggested for tuning of the K-means for the text document clustering problem. The elementary SCA is improved by the mutation and crossover operators from the well-known genetic algorithm, which has the aim to improve search space exploration and make a better trade-off among exploitation and exploration procedures.

First, we tested the introduced algorithm and the original SCA on 30 standard benchmark functions from the CEC 2017 challenging collection to prove the effectiveness of the algorithm. The unconstrained benchmark simulation outcomes have been put into comparisons with other cutting-edge metaheuristics, namely IHHO, HHO, DE, GOA, GWO, MFO, MVO, PSO, and WOA. The obtained simulation outcomes and conducted comparisons confirmed the efficiency of SCA-GA.

Secondly, the SCA-GA is applied to tune K-means for text document clustering problem and evaluated on six text document datasets (Centre for Speech Technology Research (CSTR) dataset, 20Newsgroups dataset (20Newsgroups), Tr41, Tr12, Wap, and Classic4. The achieved scores of the suggested method are compared to the other metaheuristic and non-metaheuristic approaches. The comparison of the evaluation metric results verifies the competitiveness of SCA-GA performance in tuning of the K-means in text document clustering.

The limits of the work presented in this manuscript are related to the fact that the introduced method has been applied only to the datasets described in the paper, while other datasets were not evaluated. According to the NFL, each dataset is specific, and there is no general solution that would perform perfectly on every dataset that exists. Practical limitation of the proposed work is related to the solving of this task in real time that could be challenging as metaheuristics algorithms need some time to execute.

Acknowledgments

This research is supported by Ministry of Education and Science of the Republic of Serbia, Grant No. III-44006 and by the Science Fund of the Republic of Serbia, grant no. 6526093, AI–AVANTES (http://fondzanauku.gov.rs/).

Availability of Data and Material

All data generated or analyzed during this study are included in this article.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Abasi, A. K., Khader, A. T., Al-Betar, M. A., Naim, S., Alyasseri, Z. A. A., Makhadmeh, S. N. A Novel Hybrid Multi-Verse Optimizer with K-means for Text Documents Clustering. Neural Computing and Applications, 2020, 32, 17703-17729. https://doi.org/10.1007/s00521-020-04945-0
2. Abasi, A. K., Khader, A. T., Al-Betar, M. A., Naim, S., Alyasseri, Z. A. A., Makhadmeh, S. N. A Novel Hybrid Multi-Verse Optimizer with K-Means for Text Documents Clustering. Neural Computing and Applications, 2020, 32, 17703-17729. https://doi.org/10.1007/s00521-020-04945-0
3. Abualigah, L. M., Khader, A. T., Al-Betar, M. A., Awadallah, M. A. A Krill Herd Algorithm for Efficient Text Documents Clustering. In: 2016 IEEE Symposium on Computer Applications Industrial Electronics (ISCAIE), 2016, 67-72. https://doi.org/10.1109/ISCAIE.2016.7575039
4. Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., Kochut, K. J. A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques, 2017. ArXiv abs/1707.02919
5. Bacanin, N., Bezdan, T., Tuba, E., Strumberger, I., Tuba, M., Zivkovic, M. Task Scheduling in Cloud Computing Environment by Grey Wolf Optimizer. 2019 27th Telecommunications Forum (TELFOR), 2019, 1-4. https://doi.org/10.1109/TELFOR48224.2019.8971223
6. Bacanin, N., Sarac, M., Budimirovic, N., Zivkovic, M., Al Zubi, A. A., Bashir, A. K. Smart Wireless Health Care System Using Graph LSTM Pollution Prediction and Dragonfly Node Localization. Sustainable Computing:

Informatics and Systems, 2022, 35, 100711. https://doi.org/10.1016/j.suscom.2022.100711

7. Bacanin, N., Venkatachalam, K., Bezdan, T., Zivkovic, M., Abouhawwash, M. A Novel Firefly Algorithm Approach for Efficient Feature Selection with COVID-19 Dataset. Microprocessors and Microsystems, 2023, 98, 104778. https://doi.org/10.1016/j.micpro.2023.104778
8. Bacanin, N., Zivkovic, M., Bezdan, T., Venkatachalam, K., Abouhawwash, M. Modified Firefly Algorithm for Workflow Scheduling in Cloud-Edge Environment. Neural Computing and Applications, 2022, 34, 9043-9068. https://doi.org/10.1007/s00521-022-06925-y
9. Baeza-Yates, R. A., Ribeiro-Neto, B. A. Modern Information Retrieval. ACM Press / Addison-Wesley, 1999. URL: http://www.ischool.berkeley.edu/~hearst/irbook/glossary.html
10. Bezdan, T., Cvetnic, D., Gajic, L., Zivkovic, M., Strumberger, I., Bacanin, N. Feature Selection by Firefly Algorithm with Improved Initialization Strategy. 7th Conference on the Engineering of Computer Based Systems, 2021, 7, 1-8. https://doi.org/10.1145/3459960.3459974
11. Bezdan, T., Stoean, C., Naamany, A. A., Bacanin, N., Rashid, T. A., Zivkovic, M., Venkatachalam, K. Hybrid Fruit-Fly Optimization Algorithm with K-means for Text Document Clustering. Mathematics, 2021, 9, 1929. https://doi.org/10.3390/math9161929
12. Bharti, K. K., Singh, P. K. Chaotic Gradient Artificial Bee Colony for Text Clustering. Soft Computing, 2016, 20, 1113-1126. https://doi.org/10.1007/s00500-014-1571-7
13. Bukumira, M., Antonijevic, M., Jovanovic, D., Zivkovic, M., Mladenovic, D., Kunjadic, G. Carrot Grading System Using Computer Vision Feature Parameters and a Cascaded Graph Convolutional Neural Network. Journal of Electronic Imaging, 2022, 31, 1-16. https://doi.org/10.1117/1.JEI.31.6.061815
14. Chandran, T., Reddy, A., Janet, B. A Social Spider Optimization Approach for Clustering Text Documents. In: 2016 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), 2016, 22-26. https://doi.org/10.1109/AEEICB.2016.7538275
15. Chen, C.-H. Improved TFIDF in Big News Retrieval: An Empirical Study. Pattern Recognition Letters, 2017, 93,113-122. https://doi.org/10.1016/j.patrec.2016.11.004
16. Chen, H.-N., He, B., Yan, L., Li, J., Ji, W. A Text Clustering Method Based on Two-Dimensional Otsu and PSO Algorithm. In: 2009 International Symposium on Computer Network and Multimedia Technology, 2009, 1-4. https://doi.org/10.1109/CNMT.2009.5374525
17. Chopard, B., Tomassini, M. Performance and Limitations of Metaheuristics. In: An Introduction to Metaheuristics for Optimization, 2018, 191-203. https://doi.org/10.1007/978-3-319-93073-2_11
18. Cuk, A., Bezdan, T., Bacanin, N., Zivkovic, M., Venkatachalam, K., Rashid, T. A., Devi, V. K. Feedforward Multi-Layer Perceptron Training by Hybridized Method between Genetic Algorithm and Artificial Bee Colony. In: Data Science and Data Analytics, Chapman and Hall/CRC, 2021, 279-292. https://doi.org/10.1201/9781003111290-17-21
19. Figueiredo, E. M. N., Macedo, M., Siqueira, H. V., Santana, C. J., Gokhale, A. A., Filho, C. J. A. B. Swarm Intelligence for Clustering - A Systematic Review with New Perspectives on Data Mining. Engineering Applications of Artificial Intelligence, 2019, 82, 313-329. https://doi.org/10.1016/j.engappai.2019.04.007
20. Gupta, S., Deep, K. Improved Sine Cosine Algorithm with Crossover Scheme for Global Optimization. Knowledge-Based Systems, 2019, 165, 374-406. https://doi.org/10.1016/j.knosys.2018.12.008
21. Holland, J. H. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. MIT Press, 1992. https://doi.org/10.7551/mitpress/1090.001.0001
22. Hussien, A. G., Amin, M. A Self-Adaptive Harris Hawks Optimization Algorithm with Opposition-Based Learning and Chaotic Local Search Strategy for Global Optimization and Feature Selection. International Journal of Machine Learning and Cybernetics, 2021, 1-28. https://doi.org/10.1007/s13042-021-01326-4
23. Yao, J., Wan, X., Xiao, J. Recent Advances in Document Summarization. Knowledge and Information Systems, 2017, 53, 297-336. https://doi.org/10.1007/s10115-017-1042-4
24. Jain, V., Shrivastava, N. Class Based Clustering with Cuckoo Search Rank Optimization for Text Data Categorization. International Journal of Master of Engineering Research and Technology, 2015, 2, 82-87.
25. Jensi, R., Jiji, G. W. A Survey on Optimization Approaches to Text Document Clustering, 2014. ArXiv abs/1401.2229
26. Jovanovic, D., Antonijevic, M., Stankovic, M., Zivkovic, M., Tanaskovic, M., Bacanin, N. Tuning Machine Learn-

ing Models Using a Group Search Firefly Algorithm for Credit Card Fraud Detection. Mathematics, 2022, 10, 2272. https://doi.org/10.3390/math10132272

27. Jovanovic, L., Jovanovic, D., Bacanin, N., Jovancai Stakic, A., Antonijevic, M., Magd, H., Thirumalaisamy, R., Zivkovic, M. Multi-Step Crude Oil Price Prediction Based on LSTM Approach Tuned by Salp Swarm Algorithm with Disputation Operator. Sustainability, 2022, 14, 14616. https://doi.org/10.3390/su142114616
28. Jovanovic, L., Jovanovic, G., Perisic, M., Alimpic, F., Stanisic, S., Bacanin, N., Zivkovic, M., Stojic, A. The Explainable Potential of Coupling Metaheuristics-Optimized-XGBoost and SHAP in Revealing VOCs' Environmental Fate. Atmosphere, 2023, 14, 109. https://doi.org/10.3390/atmos14010109
29. Kalogeratos, A., Likas, A. Text Document Clustering Using Global Term Context Vectors. Knowledge and Information Systems, 2011, 31, 455-474. https://doi.org/10.1007/s10115-011-0412-6
30. Karaboga, D., Ozturk, C. A Novel Clustering Approach: Artificial Bee Colony (ABC) Algorithm. Applied Soft Computing, 2011, 11, 652-657. https://doi.org/10.1016/j.asoc.2009.12.025
31. Lovins, J. B. Development of a Stemming Algorithm. Mechanical Translation and Computational Linguistics, 1968, 11, 22-31
32. Mirjalili, S. SCA: A Sine Cosine Algorithm for Solving Optimization Problems. Knowledge-Based Systems, 2016, 96, 120-133. https://doi.org/10.1016/j.knosys.2015.12.022
33. Nanda, S. J., Panda, G. A Survey on Nature Inspired Metaheuristic Algorithms for Partitional Clustering. Swarm and Evolutionary Computation, 2014, 16, 1-18. https://doi.org/10.1016/j.swevo.2013.11.003
34. Ponnusamy, M., Bedi, P., Suresh, T., Alagarsamy, A., Manikandan, R., Yuvraj, N. Design and Analysis of Text Document Clustering Using Salp Swarm Algorithm. The Journal of Supercomputing, 2022, 78, 16197-16213. https://doi.org/10.1007/s11227-022-04525-0
35. Porter, M. F. An Algorithm for Suffix Stripping. Program, 40, 1980, 211-218. https://doi.org/10.1108/00330330610681286
36. Purushothaman, R., Rajagopalan, S., Dhandapani, G. Hybridizing Gray Wolf Optimization (GWO) with Grasshopper Optimization Algorithm (GOA) for Text Feature Selection and Clustering. Applied Soft Computing, 2020, 96, 106651. https://doi.org/10.1016/j.asoc.2020.106651
37. Rana, S., Jasola, S., Kumar, R. A Review on Particle Swarm Optimization Algorithms and Their Applications to Data Clustering. Artificial Intelligence Review, 2010, 35, 211-222. https://doi.org/10.1007/s10462-010-9191-9
38. Rezaeiye, P. P., Bazrafkan, M., Movassagh, A. A., Fazli, M. S., Bazyari, G. H. Use HMM and KNN for Classifying Corneal Data, 2014. arXiv:1401.7486
39. Sarac, M., Mravik, M., Jovanovic, D., Strumberger, I., Zivkovic, M., Bacanin, N. Intelligent Diagnosis of Coronavirus with Computed Tomography Images Using a Deep Learning Model. Journal of Electronic Imaging, 32, 021406 (2022). https://doi.org/10.1117/1.JEI.32.2.021406
40. Song, W., Li, C. H., Park, S. C. Genetic Algorithm for Text Clustering Using Ontology and Evaluating the Validity of Various Semantic Similarity Measures. Expert Systems with Applications, 2009, 36, 9095-9104. https://doi.org/10.1016/j.eswa.2008.12.046
41. Stoean, C., Zivkovic, M., Bozovic, A., Bacanin, N., Strulak-Wojcikiewicz, R., Antonijevic, M., Stoean, R. Metaheuristic-Based Hyperparameter Tuning for Recurrent Deep Learning: Application to the Prediction of Solar Energy Generation. Axioms, 2023, 12, 266. https://doi.org/10.3390/axioms12030266
42. Strumberger, I., Minovic, M., Tuba, M., Bacanin, N. Performance of Elephant Herding Optimization and Tree Growth Algorithm Adapted for Node Localization in Wireless Sensor Networks. Sensors (Basel, Switzerland), 2019, 19, 1-18. https://doi.org/10.3390/s19112515
43. Strumberger, I., Tuba, E., Bacanin, N., Zivkovic, M., Beko, M., Tuba, M. Designing Convolutional Neural Network Architecture by the Firefly Algorithm. 2019 International Young Engineers Forum (YEF-ECE), 2019, 59-65. https://doi.org/10.1109/YEF-ECE.2019.8740818
44. Sutton, A., Clowes, M., Preston, L., Booth, A. Meeting the Review Family: Exploring Review Types and Associated Information Retrieval Requirements. Health Information and Libraries Journal, 2019, 36(3), 202-222. https://doi.org/10.1111/hir.12276
45. Tomer, M., Kumar, M. Multi-Document Extractive Text Summarization Based on Firefly Algorithm. Journal of King Saud University-Computer and Information Sciences, 2022, 34, 6057-6065. https://doi.org/10.1016/j.jksuci.2021.04.004
46. Webster, J. J., Kit, C. Tokenization as the Initial Phase in NLP. In Proceedings of the 14th Conference on Computational Linguistics - Volume 4, COLING'92, Associ-

ation for Computational Linguistics, USA, 1992, 1106-1110. https://doi.org/10.3115/992424.992434

47. Wolpert, D. H., Macready, W. G. No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation, 1997, 1, 67-82. https://doi.org/10.1109/4235.585893
48. Wu, G., Mallipeddi, R., Suganthan, P. N. Problem Definitions and Evaluation Criteria for the CEC. 2017 Competition on Constrained Real-Parameter Optimization, 2017.
49. Zivkovic, M., Bacanin, N., Antonijevic, M., Nikolic, B., Kvascev, G., Marjanovic, M., Savanovic, N. Hybrid CNN and XGBoost Model Tuned by Modified Arithmetic Optimization Algorithm for COVID-19 Early Diagnostics from X-ray Images. Electronics, 2022, 11, 3798. https://doi.org/10.3390/electronics11223798
50. Zivkovic, M., Bacanin, N., Venkatachalam, K., Nayyar, A., Djordjevic, A., Strumberger, I., Al-Turjman, F. Covid-19 Cases Prediction by Using Hybrid Machine Learning and Beetle Antennae Search Approach. Sustainable Cities and Society, 2021, 66, 102669. https://doi.org/10.1016/j.scs.2020.102669
51. Zivkovic, M., Bezdan, T., Strumberger, I., Bacanin, N., Venkatachalam, K. Improved Harris Hawks Optimization Algorithm for Workflow Scheduling Challenge in Cloud-Edge Environment. In: Computer Networks, Big Data and IoT: Proceedings of ICCBI, Springer, 2021, 87-102. https://doi.org/10.1007/978-981-16-0965-7_9
52. Zivkovic, M., Tair, M., Venkatachalam, K., Bacanin, N., Hubalovsky, S., Trojovsky, P. Novel Hybrid Firefly Algorithm: An Application to Enhance XGBoost Tuning for Intrusion Detection Classification. PeerJ Computer Science, 2022, 8, e956. https://doi.org/10.7717/peerj-cs.956