Article

Hybrid Fruit-Fly Optimization Algorithm with K-Means for Text Document Clustering

Timea Bezdan ¹ , Catalin Stoean ² , Ahmed Al Naamany ³ , Nebojsa Bacanin ^{1,*} , Tarik A. Rashid ⁴ ,
Miodrag Zivkovic ¹ and K. Venkatachalam ⁵

¹ Faculty of Informatics and Computing, Singidunum University, Danijelova 32, 11010 Belgrade, Serbia; tbezdan@singidunum.ac.rs (T.B.), mzivkovic@singidunum.ac.rs (M.Z.)

² Human Language Technology Research Center, University of Bucharest, 010014 Bucharest, Romania; catalin.stoean@fmi.unibuc.ro

³ Department for Mathematics and Computer Science, Modern College of Business and Science, Muscat 113, Oman; naamany@mcbs.edu.om

⁴ Computer Science and Engineering Department, University of Kurdistan Hewler, Erbil 44001, Iraq; tarik.ahmed@ukh.edu.krd

⁵ Department of Computer Science and Engineering, CHRIST (Deemed to be University), Bangalore 560029, India; venkatachalam.k@christuniversity.in

* Correspondence: nbacanin@singidunum.ac.rs; Tel.: +381-653093-224

Abstract: The fast-growing Internet results in massive amounts of text data. Due to the large volume of the unstructured format of text data, extracting relevant information and its analysis becomes very challenging. Text document clustering is a text-mining process that partitions the set of text-based documents into mutually exclusive clusters in such a way that documents within the same group are similar to each other, while documents from different clusters differ based on the content. One of the biggest challenges in text clustering is partitioning the collection of text data by measuring the relevance of the content in the documents. Addressing this issue, in this work a hybrid swarm intelligence algorithm with a K-means algorithm is proposed for text clustering. First, the hybrid fruit-fly optimization algorithm is tested on ten unconstrained CEC2019 benchmark functions. Next, the proposed method is evaluated on six standard benchmark text datasets. The experimental evaluation on the unconstrained functions, as well as on text-based documents, indicated that the proposed approach is robust and superior to other state-of-the-art methods.

Keywords: machine learning; text document clustering; metaheuristic algorithms; fruit-fly optimization algorithm; K-means

Citation: Bezdan, T.; Stoean C.; Namany, A.A.; Bacanin, N.; Rashid, A.T.; 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>

Academic Editor: David Greiner

Received: 20 July 2021

Accepted: 30 July 2021

Published: 13 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Text document clustering has become an important and fast-growing research area, due to the massive amounts of text data produced by the Internet, social media, email and text messages, and other sources. Text-mining techniques are used to process and analyze text-based data, and by the use of different text-mining methods, relevant information can be extracted from the text data. One crucial method in text-mining is clustering, which has the aim of automatically partition the number of documents in a finite set of homogeneous clusters (groups). In a specific cluster, all documents are similar to each other based on the content, while in different clusters, the similarity decreases. From the perspective of optimization, clustering can be presented as an NP-hard optimization problem.

Metaheuristic algorithms are shown to be very efficient to solve NP-hard optimization problems and result in close-optimal solutions in a fair amount of time. Metaheuristic algorithms that are inspired by the nature can be divided into two major categories, swarm intelligence and evolutionary algorithms. In this work, a hybrid swarm intelligence algorithm is proposed. The fruit-fly optimization algorithm [1,2] is hybridized by the firefly algorithm (FA) [3]. Additionally, the opposition-based learning mechanism is incorporated

in the hybrid method, it is combined with the traditional K-means algorithm [4], and employed for text-based document clustering.

According to [5], the traditional algorithms for clustering can be grouped into 9 classes, as illustrated in Table 1. One of the main disadvantages of traditional clustering algorithms is that they can easily be drawn into local optima [5]. Swarm intelligence algorithms are shown to be very effective in avoiding the local optima issue [6]. Swarm intelligence systems generally consist of a collaborative population of individuals, who are cooperating locally with one another, as well as interacting with their environment. Each individual in the population follows simple rules and collectively produces behavior that leads together to the common goal. The random local and cooperation among the individuals lead to the evolution of intelligent global performance. This intelligent collective behavior is called swarm intelligence (SI).

Table 1. Categories of traditional algorithms for clustering and examples of each type.

Category	Examples
Partition	K-means [4], K-medoids [7], PAM [8], CLARA [8], CLARANS [9]
Hierarchy	BIRCH [10], CURE [11], ROCK [12], Chameleon [13]
Fuzzy theory	FCM [14], FCS [15], MM [16]
Distribution	DBCLASD [17], GMM [18]
Density	DBSCAN [19], OPTICS [20], Mean-shift [21]
Graph theory	CLICK [22], MST [23]
Grid	STING [24], CLIQUE [25]
Fractal theory	FC [26]
Model	COBWEB [27]

Metaheuristics in general, and swarm intelligence algorithms in particular, have many successful applications in different fields, such as image segmentation, classification, clustering [28–32], feature selection [33], data clustering [34], convolutional neural network architecture optimization [35–39], in wireless sensor networks (WSNs) [40,41], cloud computing applications [42–47], vehicle routing [48,49], path planning [50], packing problems [51], resource allocation [52], optimizations in different deep learning applications and neural network training [53–57], supply chain management [58], and many others. The fruit-fly optimization algorithm was recently successfully applied for clustering [59], although not for the specific problem of document clustering, and it was not hybridized with the FA.

Many researchers applied metaheuristic-based (including swarm intelligence) algorithms for text document clustering problems. The well-known particle swarm optimization algorithm (PSO) hybridized with K-means algorithm and OTSU algorithm is used for text clustering in [60]. In [61] grey wolf optimization algorithm (GWO) is hybridized with grasshopper optimization algorithm (GOA), and Fuzzy c-means (FCM) for feature selection and text clustering. The FA is used in [62] for dynamic document clustering. The krill herd algorithm (KH) is hybridized with genetic operators and applied for web text documents clustering in [63]. A hybrid multi-verse optimizer (MVO) with the traditional K-means clustering algorithm is used for text clustering in [64]. Text document clustering represents an important machine learning task, and our intention was to examine the performance of FFO for this application, as well as specifically evaluating the newly proposed technique in this scenario. In such tasks there is also space for improvement using other novel, improved metaheuristics and algorithms, often obtained through hybridization. Notwithstanding the existing metaheuristic-based text clustering applications, there is still a need to develop an algorithm that is more robust and efficient than the existing ones.

The following research questions inspired this work: (i) How to develop an efficient text clustering method that will group closely related documents and minimize the similarity in different clusters? (ii) How to develop an efficient swarm intelligence-based algorithm for text document clustering? The objective of this paper was to answer the research questions as follows: (i) Develop an improved hybrid fruit-fly optimization (FFO) swarm intelligence method in which performance in terms of quality of solutions is superior to the original algorithm. (ii) Employ the novel hybrid FFO approach to text document clustering and produce better results as compared to the ones obtained by other state-of-the-art approaches for the same problem.

The rest of the paper is organized as follows: the text document clustering problem formulation is described in Section 2 and the explanation of the proposed method is given in Section 3. Section 4 demonstrates the simulation results for the optimization of unconstrained benchmark functions, as well as it presents the text datasets used for evaluation purposes, evaluation metrics, and simulation results along with the comparative analysis. Finally, Section 5 concludes the paper.

2. Text Document Clustering

Text Document Clustering (TDC) represents the process of partitioning a set of heterogeneous documents into distinctive clusters based upon the content similarity. Before clustering technique application, data pre-processing should be performed. The pre-processing steps in text-based documents are tokenization, stop words are removed, stemming, lemmatization and, as a last step, conversion of the text to numerical form are performed.

2.1. Tokenization

In the tokenization process of text-based documents, each document in the corpus (collection of all documents) is converted to a list of tokens (units). In the case of a sentence, the token would be a word, while in the case of a word input, the tokens would be characters.

For example, having the following sentence:

input_sentence = An example of tokenization.

The list of tokens after the tokenization process would be as follows:

token_list = (An, example, of, tokenization)

2.2. Stop Words Removal

In the next process of text pre-processing, the common words, such as articles, conjunctions, etc. are removed from the list of tokens. By applying the stop words removal process on the above example, the result would be as follows:

filtered_tokens = (example, tokenization)

where *an* and *of* are stop words and they are removed from the token list.

Stop words removal reduces the size of the vocabulary, which leads to faster computation.

2.3. Stemming and Lemmatization

In the process of stemming and lemmatization, the words are normalized, meaning that the words are reduced to their root form. The difference between stemming and lemmatization is that in stemming, the last few characters are removed from the word, while in lemmatization, the context is taken into account, and the word is converted to its base form (lemma). For example, applying the stemming process on word *studies*, would result in *studi*, where the *es* are removed from the end of the word, while in the case of lemmatization, the result would be *study*, which is the actual lemma of the word.

2.4. Vectorization

The last step in the text pre-processing pipeline is converting the words to their numerical forms for building the vocabulary that contains all tokens; this process is called vectorization. There are different vectorization techniques, the Term Frequency-Inverse Document Frequency (TF-IDF) is one of the most commonly used approaches.

The i -th document is represented by the weight vector as described in Equation (1).

$$d_i = \{w_{i,1}, w_{i,2}, w_{i,3}, \dots, w_{i,j}, \dots, w_{i,t}\} \quad (1)$$

The weight of document terms is calculated according to the formula in Equation (2), where i represents a term, and j represents a specific document. $TF(i, j)$ denotes the frequency of i -th term in document j , $DF(i)$ refers to the number of documents that contains the term i , while d is the total number of documents.

$$w_{i,j} = TF - IDF(i, j) = TF(i, j) \times \left(\log \frac{d}{DF(i)}\right) \quad (2)$$

The vector space model (VSM) is presented in Equation (3).

$$VSM = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,t-1} & w_{1,t} \\ \dots & \dots & \dots & \dots & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{n-1,1} & w_{n-1,2} & \dots & w_{n-1,t-1} & w_{n-1,t} \\ w_{n,1} & w_{n,2} & \dots & w_{n,t-1} & w_{n,t} \end{bmatrix} \quad (3)$$

3. Proposed Method

This section, first introduces the original FFO algorithm, points out its advantages and drawbacks. Subsequently, the proposed hybrid method is described, followed by the adaptation of the proposed method for text document clustering problem.

3.1. Original Fruit-Fly Optimization Algorithm

The original fruit-fly optimization (FFO) algorithm is introduced by Pan [1,2]. The FFO is inspired by the fruit-fly's foraging behavior. The original fruit-fly optimization algorithm consists of four phases:

- initialization,
- osphresis foraging,
- population evaluation and
- vision.

Initially, the solutions (fruit flies) are generated randomly within the given lower and upper bounds, according to Equation (4), where $x_{i,j}$ denotes i -th solution, and j subscript indicates to the element's position in the i -th solution. lb , and ub refers to the lower and upper bound, respectively. $rand$ is a random number drawn from the uniform distribution.

$$x_{i,j} = rand(ub_j - lb_j) + lb_j \quad (4)$$

After the population is initialized, the position of each solution is updated according to the osphresis foraging phase, where the solution moves a random step from the current location, and it is formulated mathematically as in Equation (5), where the new position is denoted by $x_{i,j}^{(t+1)}$, the current solution is $x_{i,j}^{(t)}$, $rand() \in [-1, 1]$. The iteration counter is denoted by t .

$$x_{i,j}^{(t+1)} = x_{i,j}^{(t)} \pm rand() \quad (5)$$

After the position update, the fitness value is calculated for each solution, then the greedy selection procedure will decide whether the new position will remain or the old one will be kept. This phase of the algorithm is known as vision foraging phase. If the new solution's fitness value is better than the old one, the new solution will replace the old, otherwise, the old solution will remain in the population and the new one will be discarded. The algorithm is terminated upon reaching the stop criteria and it returns the best solution.

3.2. Proposed Hybrid FFO

The FFO's structure is relatively simple, it has a reduced number of parameters, and the adaptation of the algorithm to a specific problem is straightforward. However, besides these advantages, the method has some drawbacks. The algorithm can become trapped in the local minima, its position update strategy is fixed, and it is poor in the exploitation process. The deficiencies of the algorithm were observed during experiments on unconstrained functions.

For the sake of enhancing the exploitation, the firefly algorithm (FA) search mechanism is incorporated in the algorithm, which excels at intensification. Additionally, to explore better the search space, opposition-based learning (OBL) is introduced.

The hybrid algorithm is named hybrid enhanced fruit-fly firefly, in short HEFFF. Initially, the population is generated randomly according to Equation (4). In even iterations, the FFO search mechanism is used if a random number is less than 0.5, otherwise, the opposition-based learning is applied, while in every odd iteration, the firefly [3] search mechanism is employed.

The opposite number (X') is defined as shown in Equation (6).

$$x_i^{(t+1)} = lb + ub - x_i^{(t)} \quad (6)$$

The firefly search mechanism updates the position by equation Equation (7), where the distance between the current candidate solution x_i and another one x_j is denoted by $r_{i,j}$ and it is calculated by Equation (8). The attractiveness, at distance zero, toward the fittest firefly (solution) is denoted by β_0 . γ and alpha are control parameters for randomization, and κ is generated from Gaussian distribution.

$$x_i^{(t+1)} = x_i^{(t)} + \frac{\beta_0}{1 + \gamma r_{i,j}^2} (x_j - x_i) + \alpha(\kappa - 0.5) \quad (7)$$

$$r_{i,j} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^D (x_{i,k} - x_{j,k})^2} \quad (8)$$

The reader may refer [3] for more details about the original FA algorithm.

For further improving the performance of the algorithm, a dynamic step is employed for the α control parameter according to [65], where the α value is decreasing gradually from the initial value (α_0), until the defined minimum value (α_{min}) throughout the iterations. The update of the α value in each iteration is defined as in Equation (9), where the current iteration is denoted by t , $MaxIter$ refers to the maximum number of iterations, $\alpha^{(t)}$ denotes the value of α at the current iteration, while the updated value is denoted by $\alpha^{(t+1)}$.

$$\alpha^{(t+1)} = \alpha^{(t)} \cdot \left(1 - \frac{t}{MaxIter}\right) \quad (9)$$

Algorithm 1 illustrates the main steps of the proposed HEFFF approach.

Algorithm 1 Pseudo-code of proposed HEFFF

```

Initialize the population randomly by Equation (4)
Initialize the FA parameters of  $\beta_0, \gamma, \alpha$ 
Set the iteration counter  $t$  to 0 and define the termination criteria
Evaluate the fitness of each individuals
while termination criteria is not satisfied do
  for  $i = 1$  to  $N$  do
    if  $t$  is even then
      if  $rand < 0.5$  then
        Update the position according to FFO updating mechanism by Equation (5)
      else
        Use OBL procedure by Equation (6)
      end if
    else
      Perform FA search using Equation (7)
    end if
  end for
  Evaluate the solutions, and if the new solution has better fitness than the old one, replace the old solution
  Sort the population according to the fitness value of the solutions and save the best solution
  Replace the worst solutions (30% in the population) by new random solutions
end while
Return the best solution

```

3.3. Hybrid FFO Adapted for Text Clustering

In the proposed hybrid FFO for the text clustering problem, the K-means procedure is incorporated as an intermediate step. Within pre-processing step of the text data, the features are normalized between 0 and 1, thus, the lower and upper bounds are also 0 and 1. Initially, the solutions are generated randomly between the lower and upper bounds.

One individual in the population represents one possible solution for clustering the text corpus data. Hence, the population represents the candidate solutions for the text corpus data clustering. Each solution is encoded as a matrix $X_i = (C_1, C_2, \dots, C_i, \dots, C_k)$, where C_i represents the i -th cluster centroid vector and k is the number of clusters. Based on the distance measure, each document should be assigned to the closest cluster.

The distance between a document and centroid is calculated as in Equation (10), where t refers to the number of terms (features) in a document, d denotes the document, and c is the centroid.

$$dist(d_j, c_k) = \sqrt{\sum_{i=1}^t |d_{ij} - c_{ik}|^2} \quad (10)$$

The fitness value is based on the K-means algorithm, and it represents the average distance of documents to the cluster centroid. The fitness value is measured by the equation Equation (11), where K denotes the total number of clusters, the number of documents in i -th cluster is denoted by n_i , the j -th document in cluster i is denoted by d_{ij} , the centroid of i -th cluster is denoted by c_i . The $dist()$ function calculates the distance between the document d_{ij} and cluster c_i .

$$f(x) = \frac{\sum_{i=1}^K (\frac{1}{n_i} \sum_j^{n_i} dist(d_{ij}, c_i))}{K} \quad (11)$$

Based on the position updating mechanisms in the proposed method, the values of the centroid vector are adjusted in every iteration, then the distance from the documents to the centroids are recalculated and reassigned. This procedure is repeated until the termination criteria are met.

4. Experimental Results

The proposed method is first validated on unconstrained benchmark functions, next it is applied for TDC. The experimental setup, problem sets, and the obtained results are presented in the following two subsections.

4.1. Unconstrained Experiment on CEC 2019 Test Suit

The performance of the proposed method is validated on 10 modern CEC2019 functions [66] and the results are compared to the original FFO, and other nine metaheuristic-based approaches (EHOI, EHO, SCA, SSA, GOA, WOA, BBO, MFO, PSO) [67], where the simulations were conducted under similar condition and the same problem sets are used.

The population size is set to 50, the maximum number of iterations is 500. The experiment is repeated in 30 independent runs. The control parameters of the unconstrained benchmark function experiment are shown in Table 2.

Table 2. Control parameter summary.

Parameter Description	Notation	Value
Population size	NP	50
Maximum iteration number	$MaxIter$	500
Absorption coefficient	γ	1
Attractiveness parameter at $r = 0$	β_0	1
Randomization parameter	α	changes according to Equation (9)
Initial α value	α_0	0.5

The obtained experimental results and the comparison of the mean value and standard deviation of the proposed method and counterparts are presented in Table 3. The results of EHOI, EHO, SCA, SSA, GOA, WOA, BBO, MFO, and PSO are taken from [67].

Table 3. Result comparison of different approaches on CEC2019 benchmark functions.

Function	Stats	Techniques										
		EHOI	EHO	SCA	SSA	GOA	WOA	BBO	MFO	PSO	FFO	HEFFF
CEC01	mean	4.7×10^4	1.4×10^7	9.5×10^9	2.7×10^9	1.5×10^{10}	1.1×10^{10}	3.4×10^{10}	6.6×10^9	8.7×10^{11}	1.5×10^3	6.8×10^3
	std	2.9×10^3	7.8×10^6	7.9×10^9	2.6×10^9	3.1×10^{10}	8.7×10^9	2.6×10^{10}	8.5×10^9	9.2×10^{11}	4.5×10^2	2.1×10^2
CEC02	mean	1.7×10^1	1.7×10^1	1.7×10^1	1.7×10^1	1.7×10^1	1.7×10^1	9.2×10^1	1.7×10^1	1.0×10^4	2.8×10^0	1.5×10^0
	std	1.2×10^{-15}	4.6×10^{-15}	4.2×10^{-2}	8.0×10^{-5}	1.5×10^{-2}	2.8×10^{-3}	2.7×10^1	3.7×10^{-15}	3.8×10^3	1.4×10^2	9.6×10^1
CEC03	mean	1.3×10^1	1.3×10^1	1.3×10^1	1.3×10^1	1.3×10^1	1.3×10^1	1.3×10^1	1.3×10^1	1.3×10^1	9.7×10^0	9.7×10^0
	std	1.9×10^{-15}	1.9×10^{-15}	1.0×10^{-4}	2.4×10^{-15}	1.2×10^{-4}	1.4×10^{-7}	2.6×10^{-7}	3.5×10^{-5}	6.5×10^{-4}	4.3×10^{-1}	5.0×10^{-1}
CEC04	mean	1.3×10^1	1.5×10^1	1.1×10^3	3.4×10^1	1.5×10^2	3.1×10^2	7.8×10^1	1.3×10^2	7.3×10^1	4.0×10^0	1.0×10^0
	std	3.9×10^0	6.3×10^0	3.9×10^2	1.2×10^1	2.0×10^2	1.2×10^2	2.6×10^1	1.8×10^2	7.8×10^0	4.3×10^{-1}	1.8×10^0
CEC05	mean	1.0×10^0	1.0×10^0	2.2×10^0	1.2×10^0	1.3×10^0	1.6×10^0	1.3×10^0	1.1×10^0	1.5×10^0	1.0×10^0	1.0×10^0
	std	2.1×10^{-2}	2.2×10^{-2}	7.6×10^{-2}	1.1×10^{-1}	1.2×10^{-1}	4.0×10^{-1}	9.8×10^{-2}	8.0×10^{-2}	1.2×10^{-1}	2.4×10^{-2}	2.4×10^{-2}
CEC06	mean	8.3×10^0	9.5×10^0	1.1×10^1	3.7×10^0	6.2×10^0	9.0×10^0	5.8×10^0	5.3×10^0	1.1×10^1	1.6×10^0	1.3×10^0
	std	8.2×10^{-1}	1.3×10^0	7.4×10^{-1}	1.4×10^0	1.3×10^0	1.1×10^0	6.5×10^{-1}	2.2×10^0	6.7×10^{-1}	6.2×10^{-2}	4.6×10^{-2}
CEC07	mean	1.4×10^2	1.8×10^2	6.6×10^2	2.9×10^2	3.0×10^2	4.5×10^2	4.8×10^0	3.2×10^2	6.1×10^2	5.9×10^0	3.9×10^0
	std	1.0×10^2	1.5×10^2	1.5×10^2	2.3×10^2	1.7×10^2	2.2×10^2	1.3×10^2	2.1×10^2	1.6×10^2	1.1×10^2	9.2×10^1
CEC08	mean	2.7×10^0	2.8×10^0	6.0×10^0	5.2×10^0	5.5×10^0	5.8×10^0	4.6×10^0	5.7×10^0	5.1×10^0	1.1×10^0	1.6×10^0
	std	8.8×10^{-1}	1.1×10^0	5.4×10^{-1}	6.3×10^{-1}	8.0×10^{-1}	7.9×10^{-1}	1.1×10^0	5.8×10^{-1}	7.4×10^{-1}	8.0×10^{-1}	5.2×10^{-1}
CEC09	mean	2.3×10^0	2.4×10^0	1.0×10^2	2.4×10^0	2.5×10^0	4.7×10^0	3.5×10^0	2.5×10^0	2.9×10^0	3.9×10^0	1.9×10^0
	std	6.2×10^{-3}	1.3×10^{-2}	9.3×10^1	4.5×10^{-2}	7.2×10^{-2}	7.8×10^{-1}	2.3×10^{-1}	6.0×10^{-2}	9.7×10^{-2}	2.8×10^{-1}	1.0×10^{-2}
CEC10	mean	2.0×10^1	2.0×10^1	2.0×10^1	2.0×10^1	2.0×10^1	2.0×10^1	2.0×10^1	2.0×10^1	2.0×10^1	2.1×10^1	1.1×10^0
	std	1.5×10^0	9.8×10^{-2}	8.1×10^{-2}	8.3×10^{-2}	9.1×10^{-2}	4.9×10^{-2}	2.4×10^{-2}	1.5×10^{-1}	1.0×10^{-1}	9.1×10^{-6}	1.0×10^{-2}

The results in Table 3 indicate the superiority of HEFFF, it has the best mean value in the case of 8 functions, while the original FFO achieved the best mean results on 2 functions (CEC01, CEC08).

Furthermore, the Friedman test [68,69] two-way analysis of variance by ranks is performed to prove the significant difference between the proposed approach and other comparable methods. The ranking of 11 algorithms when applied to the 10 functions is provided in Table 4.

Table 4. Friedman ranks of the comparable method over 10 CEC2019 benchmark functions.

Function	Techniques										
	EHOI	EHO	SCA	SSA	GOA	WOA	BBO	MFO	PSO	FFO	HEFFF
CEC01	3	4	7	5	9	8	10	6	11	1	2
CEC02	5	5	9	5	8	5	10	5	11	2	1
CEC03	7	7	7	7	7	7	7	7	7	2	1
CEC04	3	4	11	5	9	10	7	8	6	2	1
CEC05	3.5	3.5	11	6	8	10	7	5	9	2	1
CEC06	7	9	11	3	6	8	5	4	10	2	1
CEC07	4	5	11	6	7	9	3	8	10	2	1
CEC08	3	4	11	7	8	10	5	9	6	1	2
CEC09	2	3	11	4	5	10	8	6	7	9	1
CEC10	2	8	10	3	4.5	6.5	4.5	6.5	9	11	1
Average	3.95	5.25	9.9	5.1	7.15	8.35	6.65	6.45	8.6	3.4	1.2
Rank	3	5	11	4	8	9	7	6	10	2	1

Table 4 shows that HEFFF is superior to the other 10 algorithms with an average ranking of 1.2 for the Friedman test. The Friedman statistics ($\chi_r^2 = 58.9$) is greater than the χ^2 critical value with 10 degrees of freedom (18.3), at significance level $\alpha = 0.005$, thus the null hypothesis (H_0) is rejected and we can conclude that HEFFF is significantly different from the other 10 methods. As reported in [70], the Iman and Davenport's test [71] may be more accurate than the approximation of chi-square; for that reason, we also conducted the Iman and Davenport's test. The obtained statistic of 1.29×10 , which is greater than the F -distribution critical value ($F(9, 9 \times 10) = 1.94$). Consequently, the second test also rejects H_0 . In the case of both tests, the p -value is less than the significance level. The summary of the statistical results is provided in Table 5.

Table 5. Friedman and Iman-Davenport test results ($\alpha = 0.05$).

Friedman Value	χ^2 Critical Value	p -Value	Iman-Davenport Value	F Critical Value	p -Value
5.89×10	1.83×10	1.11×10^{-16}	1.29×10	1.94	1.11×10^{-13}

As the null hypothesis is rejected by Friedman and Iman and Davenport tests, we proceeded with the non-parametric post-hoc procedure, with the Holm's step-down procedure, and the obtained results are reported in Table 6.

Table 6. Holm's step-down procedure result.

Comparison	p -Value	Rank	$0.05/(k - i)$	$0.1/(k - i)$
SCA	2.24×10^{-9}	0	0.005000	0.01000
PSO	3.03×10^{-7}	1	0.005556	0.01111
WOA	7.16×10^{-7}	2	0.006250	0.01250
GOA	3.02×10^{-5}	3	0.007143	0.01429
BBO	1.1×10^{-4}	4	0.008333	0.01667
MFO	2.00×10^{-4}	5	0.010000	0.02000
EHO	3.16×10^{-3}	6	0.012500	0.02500
SSA	4.28×10^{-3}	7	0.016667	0.03333
EHOI	3.19×10^{-2}	8	0.025000	0.05000
FFO	6.90×10^{-2}	9	0.050000	0.10000

Table 6 shows that the proposed method significantly outperformed the compared method at significance level $\alpha = 0.1$, as well as the first eight methods at significance level $\alpha = 0.05$.

Figure 1 illustrates a head-to-head comparison between HEFFF and the second-best method with respect to the results on the 10 benchmark functions, i.e., FFO. The plots illustrate how the fitness evaluation decreases over the 500 iterations for each test function in turn.

Figure 1. Convergence graph of the 10 CEC 2019 benchmark functions as direct comparison between HEFFF and FFO.

4.2. Text Document Clustering Experiment

In the second experiment, 6 benchmark text datasets are used for the HEFFF validation: (i) The Centre for Speech Technology Research (CSTR) dataset has 299 documents belonging to four categories (artificial intelligence, theory, systems, and robotics); (ii) The 20Newsgroups dataset (20Newsgroups) contains 20 classes and for this research 3 of them are used, namely *the comp.windows.x*, *talk.politics.misc*, and *rec.autos*. Each of the three categories has 100 documents; (iii) Tr41, (iv) Tr12, and (v) Wap datasets are from Karypis lab, the number of classes by datasets are 10, 8, and 20 classes, respectively; (vi) The Classic4 dataset consists of four classes (CISI, CACM, MED, and CRAN), each class has 500 documents, in total there are 2000 documents in the dataset. The characteristics of the text datasets are described in Table 7.

Table 7. Text document clustering dataset description.

Dataset Name	Number of Documents	Number of Clusters	Number of Features
CSTR	299	4	1725
20Newsgroups	300	3	2275
tr12	313	8	5329
tr41	878	10	6743
Wap	1560	20	7512
Classic4	2000	4	6500

4.2.1. Evaluation Metrics

The error rate, accuracy, precision, recall, F-measure, purity, and entropy are used as evaluation measurements in text document clustering. Their expressions were taken from [64].

The error rate calculates the wrong cluster assignment to a document over the total number of documents in the given dataset. The mathematical formulation is shown in Equation (12), where the total number of documents in the dataset (or test dataset) is denoted by n , $n_{i,j}^-$ refers to the number of misplaced documents from class i to cluster j and k represents the number of clusters.

$$ER = \frac{1}{n} \sum_{j=1}^k n_{i,j}^- \quad (12)$$

The accuracy measure calculates the ratio of document assignment to the correct clusters (when class i is identified as cluster j) in the given dataset. The calculation is performed by equation Equation (13).

$$AC = \frac{1}{n} \sum_{j=1}^k n_{i,j} \quad (13)$$

The precision metric gives information about the correct classes over all classes in the clusters, and it is calculated as shown in Equation (14), where $n_{i,j}$ denotes the correctly assigned class i in the j cluster, and n_j indicates to the total number of documents in cluster j .

$$P(i, j) = \frac{n_{i,j}}{n_j} \quad (14)$$

The recall measures the ratio of correct document assignment over the total number of documents in the given class, at the value is determined by Equation (15), where $n_{i,j}$ denotes the correctly assigned class i in the j cluster, and n_i indicates to the total number of documents in class i .

$$R(i, j) = \frac{n_{i,j}}{n_i} \quad (15)$$

The F-measure represents the Harmonic mean of the precision and recall, and it is calculated by Equation (16), where $P(i, j)$, and $R(i, j)$ denotes the precision, and recall, respectively.

$$F(i, j) = \frac{2 \times P(i, j) \times R(i, j)}{P(i, j) + R(i, j)} \quad (16)$$

Entropy measures the distribution of documents of class labels in each cluster. The value is between 0 and 1, the best entropy is closer to 0. The entropy value of cluster j is determined by Equation (17), where $E(j)$ denotes the entropy of cluster j , the probability of class i in cluster j of a document is denoted by $p(i, j)$.

$$E(j) = - \sum_{i=1}^k p(i, j) \log p(i, j) \quad (17)$$

The entropy of clustering (all clusters) is determined by Equation (18).

$$E = \sum_{j=1}^k \frac{n_j}{n} E(j) \quad (18)$$

4.2.2. Results and Discussion

The text document clustering experiment is repeated in 30 independent runs. The maximum number of iterations is set to 1000, and the population size is 20. The control parameters are summarized in Table 8.

The text document clustering on the six text-based dataset, experimental results of the proposed method, original FFO, five non-metaheuristic methods, and 9 other metaheuristic-based methods are presented in Tables 9–14. The results of the methods used for comparison (K-mean, K-mean++, DBSCAN, Agglomerative, Spectral, KHA, HS, PSO, GA, MVO, H-PSO, H-GA, H-MVO1, H-MVO2) are taken from [64].

Table 8. Control parameter summary.

Parameter Description	Notation	Value
Population size	NP	20
Maximum iteration number	$MaxIter$	1000
Absorption coefficient	γ	1
Attractiveness parameter at $r = 0$	β_0	1
Randomization parameter	α	changes according to Equation (9)
Initial α value	α_0	0.5

As presented in the tables, the hybrid FFO method is a very competitive and promising method for text clustering. HEFFF has the best error rate in the experiment with all six datasets, and in the case of five datasets, the best accuracy is obtained by the proposed method. Moreover, in the case of five datasets, the HEFFF has the best results of recall, while the F-measure is the best on four datasets.

Table 9. Text document clustering results (CSTR dataset).

Method	ER	Accuracy	Precision	Recall	F-Measure	Purity	Entropy
K-mean	0.4029	0.3573	0.4091	0.3092	0.346	0.3525	0.8201
K-mean++	0.3678	0.4355	0.3953	0.4076	0.3546	0.4096	0.5246
DBSCAN	0.4244	0.4005	0.3393	0.4256	0.3046	0.4076	0.4586
Agglomerative	0.3672	0.436	0.4423	0.4666	0.3266	0.4816	0.5076
Spectral	0.3591	0.4319	0.3597	0.4925	0.3971	0.4485	0.4893
KHA	0.3328	0.3649	0.4213	0.5355	0.4139	0.3874	0.4344
HS	0.3431	0.4464	0.4235	0.506	0.3377	0.4355	0.4786
PSO	0.3374	0.4356	0.534	0.436	0.4819	0.4953	0.6199
GA	0.3948	0.3399	0.4417	0.3418	0.3886	0.405	0.717
MVO	0.3198	0.4593	0.5715	0.4829	0.5244	0.5685	0.5207
H-PSO	0.3028	0.5494	0.6065	0.56	0.5577	0.6135	0.5076
H-GA	0.3078	0.5056	0.613	0.495	0.5859	0.5743	0.5104
H-MVO1	0.2678	0.5683	0.6395	0.5079	0.5774	0.5015	0.4577
H-MVO2	0.2358	0.5779	0.6497	0.5568	0.5936	0.598	0.401
FFO	0.2285	0.5859	0.5757	0.5944	0.5849	0.586	0.3397
HEFFF	0.2215	0.5964	0.5873	0.5641	0.5755	0.6140	0.3361

Table 10. Text document clustering results (20News groups dataset).

Method	ER	Accuracy	Precision	Recall	F-Measure	Purity	Entropy
K-mean	0.3205	0.318	0.3121	0.31	0.3406	0.3741	0.8028
K-mean++	0.3257	0.3784	0.3652	0.3662	0.3619	0.4134	0.6611
DBSCAN	0.384	0.3038	0.3094	0.3017	0.3193	0.3027	0.7473
Agglomerative	0.3313	0.4055	0.399	0.3576	0.3548	0.4417	0.5953
Spectral	0.3143	0.3633	0.3424	0.328	0.3136	0.311	0.6125
KHA	0.2962	0.3216	0.3829	0.3136	0.2996	0.3421	0.6767
HS	0.3021	0.3122	0.3601	0.317	0.3214	0.3355	0.6481
PSO	0.2967	0.3498	0.4134	0.3497	0.3803	0.4097	0.7723
GA	0.3605	0.3676	0.4209	0.3676	0.3936	0.4081	0.7547
MVO	0.2804	0.4044	0.4392	0.3842	0.4109	0.4344	0.7121
H-PSO	0.2744	0.4442	0.4821	0.422	0.4454	0.4725	0.6631
H-GA	0.2804	0.4308	0.4814	0.4887	0.4433	0.4837	0.6833
H-MVO1	0.2864	0.5174	0.5102	0.4332	0.5169	0.5104	0.5811
H-MVO2	0.2324	0.5326	0.5619	0.4876	0.5376	0.4591	0.5407
FFO	0.2272	0.55	0.5376	0.55	0.5437	0.5100	0.3299
HEFFF	0.2238	0.5833	0.5668	0.5833	0.575	0.5100	0.3267

Table 11. Text document clustering results (tr12 dataset).

Method	ER	Accuracy	Precision	Recall	F-Measure	Purity	Entropy
K-mean	0.4384	0.2971	0.3522	0.2944	0.3222	0.3908	0.7138
K-mean++	0.4899	0.3795	0.4215	0.3778	0.4176	0.4808	0.5094
DBSCAN	0.4271	0.3045	0.3218	0.3234	0.4048	0.4728	0.501
Agglomerative	0.4422	0.4481	0.4923	0.4341	0.409	0.5553	0.3978
Spectral	0.4682	0.3373	0.4508	0.3055	0.3781	0.4132	0.4932
KHA	0.4136	0.3357	0.3748	0.3099	0.2916	0.3803	0.5183
HS	0.4484	0.3776	0.4385	0.3453	0.447	0.4986	0.4517
PSO	0.4272	0.4075	0.4298	0.4264	0.4278	0.4878	0.572
GA	0.4767	0.3677	0.4128	0.3549	0.3826	0.4513	0.6233
MVO	0.4669	0.4485	0.5075	0.4398	0.4706	0.5448	0.5224
H-PSO	0.4599	0.4796	0.5355	0.3893	0.586	0.5876	0.5757
H-GA	0.4549	0.5205	0.4798	0.4974	0.5428	0.6038	0.638
H-MVO1	0.4019	0.5465	0.6025	0.5138	0.5776	0.5948	0.4224
H-MVO2	0.4279	0.5617	0.6398	0.5109	0.5956	0.6063	0.3753
FFO	0.3859	0.5769	0.5992	0.5771	0.5879	0.535	0.6663
HEFFF	0.3685	0.6398	0.6295	0.6284	0.629	0.5744	0.6965

Table 12. Text document clustering results (tr41 dataset).

Method	ER	Accuracy	Precision	Recall	F-Measure	Purity	Entropy
K-mean	0.4459	0.4126	0.3945	0.3813	0.3876	0.4108	0.5874
K-mean++	0.4816	0.426	0.3769	0.3559	0.4299	0.5471	0.4745
DBSCAN	0.4552	0.3704	0.4242	0.4291	0.4152	0.6025	0.4776
Agglomerative	0.4405	0.461	0.3445	0.3487	0.4187	0.476	0.4821
Spectral	0.4556	0.3787	0.363	0.348	0.3691	0.4866	0.5386
KHA	0.4212	0.4054	0.4191	0.4579	0.4137	0.6063	0.4041
HS	0.3885	0.359	0.371	0.3888	0.3701	0.4863	0.4539
PSO	0.4185	0.487	0.4505	0.4497	0.4497	0.579	0.5391
GA	0.5048	0.432	0.414	0.4008	0.4071	0.5603	0.5469
MVO	0.4749	0.463	0.4569	0.4419	0.4569	0.6081	0.5355
H-PSO	0.4389	0.474	0.501	0.4768	0.4701	0.5273	0.5029
H-GA	0.4449	0.594	0.5245	0.5127	0.5647	0.642	0.5371
H-MVO1	0.4029	0.568	0.5489	0.5079	0.5119	0.6421	0.3965
H-MVO2	0.3939	0.615	0.569	0.5178	0.5201	0.6673	0.3669
FFO	0.3852	0.617	0.6111	0.6132	0.6122	0.5475	0.316
HEFFF	0.3769	0.6487	0.6341	0.642	0.638	0.5791	0.313

Table 13. Text document clustering results (Wap dataset).

Method	ER	Accuracy	Precision	Recall	F-Measure	Purity	Entropy
K-mean	0.4885	0.5012	0.4626	0.4011	0.4315	0.4759	0.7044
K-mean++	0.5145	0.4937	0.4913	0.3508	0.4462	0.5797	0.5961
DBSCAN	0.5372	0.4846	0.5004	0.4176	0.4878	0.4007	0.5706
Agglomerative	0.425	0.4933	0.4479	0.4471	0.4487	0.5445	0.4875
Spectral	0.481	0.4905	0.4454	0.3423	0.4141	0.5142	0.5468
KHA	0.4505	0.5191	0.4773	0.4106	0.4131	0.5939	0.5595
HS	0.4725	0.5032	0.463	0.4447	0.4611	0.5589	0.547
PSO	0.4598	0.5623	0.5249	0.4811	0.5017	0.6125	0.5765
GA	0.5083	0.5316	0.5314	0.4706	0.4998	0.4917	0.6216
MVO	0.4392	0.5291	0.5213	0.4496	0.4831	0.6069	0.6625
H-PSO	0.4132	0.5532	0.545	0.4907	0.5521	0.6939	0.631
H-GA	0.4272	0.6523	0.5749	0.5671	0.6187	0.7315	0.6585
H-MVO1	0.3922	0.5931	0.6063	0.5906	0.5911	0.6849	0.4365
H-MVO2	0.3762	0.6176	0.6344	0.5855	0.6105	0.7107	0.4196
FFO	0.3859	0.5769	0.5992	0.5771	0.5879	0.535	0.6663
HEFFF	0.3685	0.6398	0.6295	0.6284	0.629	0.5744	0.6965

Table 14. Text document clustering results (Classic4 dataset).

Method	ER	Accuracy	Precision	Recall	F-Measure	Purity	Entropy
K-mean	0.3461	0.5858	0.5699	0.5259	0.5472	0.5938	0.5601
K-mean++	0.3735	0.6799	0.688	0.7028	0.692	0.6227	0.354
DBSCAN	0.3951	0.6206	0.5708	0.5796	0.6173	0.6089	0.4854
Agglomerative	0.3486	0.6683	0.6449	0.6224	0.6281	0.5812	0.4033
Spectral	0.3348	0.6326	0.662	0.6528	0.6764	0.6352	0.4002
KHA	0.3031	0.5761	0.6246	0.555	0.5529	0.555	0.5321
HS	0.3166	0.5963	0.6064	0.5674	0.5787	0.5983	0.4946
PSO	0.315	0.6363	0.6604	0.6164	0.6377	0.6243	0.5306
GA	0.3694	0.6621	0.6726	0.632	0.6519	0.632	0.5781
MVO	0.2951	0.7043	0.6919	0.6844	0.6881	0.6742	0.5113
H-PSO	0.2861	0.6853	0.7354	0.6754	0.6297	0.6413	0.5656
H-GA	0.2791	0.7273	0.7774	0.6944	0.6887	0.6733	0.6166
H-MVO1	0.2461	0.7163	0.7129	0.6844	0.6951	0.7042	0.3553
H-MVO2	0.2391	0.7271	0.7226	0.705	0.7079	0.732	0.3341
FFO	0.2285	0.7296	0.6102	0.7236	0.6621	0.5501	0.262
HEFFF	0.2218	0.7459	0.6622	0.7467	0.7029	0.6311	0.288

On the CSTR dataset the HEFFF improved the accuracy by 16% over the non-metaheuristic approaches, and for 2% over the best resulted in metaheuristic method reported in [64]. HEFFF resulted in the best error rate, accuracy, precision, recall, F-measure, and entropy on the 20Newsgroups dataset, and the best purity in this dataset is obtained by H-MVO1. The best performing method on the tr12 dataset is HEFFF, with the best-obtained results for error rate, accuracy, recall, and F-measure, while the second-best approach is H-MVO2, which achieved the best result on precision, purity, and entropy. The accuracy improvement of HEFFF on tr41 is from 3% to 24% over the other methods. In the case of tr41, HEFFF has the best result on all metrics, except purity, where H-MVO2 has the best value. On the Wap dataset, the H-GA method has the best accuracy, while the proposed method achieved the best value on three metrics. On the Classic4 dataset, HEFFF improved the accuracy from 2% to 16% over other non-metaheuristic and metaheuristic approaches.

In an overall view, HEFFF is classified first in all 6 benchmark cases for the error rate and accuracy. In 5 out of 6, the proposed method has the best recall and F-measure, the only dataset where it was not best being CSTR for both measures, but where the results were still close to the best output. The best precision and entropy are obtained in 3 out of the 6 tests, while the best purity is achieved in one benchmark case.

To have an overview of the results from Tables 9–14, we additionally illustrate a metrics comparison over all methods and datasets in Figures 2 and 3.

Figure 2. Comparison by error rate, accuracy, precision and recall for all considered methods and text datasets.

Figure 3. Comparison by F-measure, purity and entropy for all considered methods and text datasets.

In both experiments, the comparison is made between our method, basic FFO and other metaheuristics. All metaheuristics are tested under similar conditions in terms of number of iterations and solutions in the population. Computational complexity of all approaches is the same in terms of number of fitness function evaluations. We kept relatively low values for the population sizes, i.e., 50 for function optimization and 20 for clustering, and had a maximum number of iterations of 500 and 1000, respectively. The running time, given these parameters, is relatively reduced, as the number of fitness evaluation calls is maintained low. Naturally, deterministic algorithms are faster to apply, but in the current experiment led to weaker results.

In setting adequate values for the parameters, we noticed that good results were obtained for a wider variety of values for the population size and the number of iterations. As concerns the other variables, they are kept the same for both types of tasks, making parameter tuning easy for the user.

Based on the experimental results, we can conclude that HEFFF is a very efficient method in text document clustering. The FFO search phase and the OBL mechanism investigates the search space globally, and by incorporating FA, the exploitation is enhanced, and it leads to efficiently finding better cluster centroids than other approaches.

5. Conclusions

In this work, a hybrid metaheuristic algorithm is proposed for text document clustering. The method is named hybrid enhanced fruit-fly firefly, for short HEFFF. The FFO algorithm is hybridized with the FA, and the OBL mechanism is additionally incorporated in the algorithm to better explore the search space on a global scale.

The method is tested on 10 unconstrained benchmark functions from CEC 2019 test suit, and the obtained results are compared to other metaheuristic techniques (EHOI, EHO, SCA, SSA, GOA, WOA, BBO, MFO, PSO). In the comparison, the fitness means to value and the standard deviation is compared between the different approaches, where the proposed method outperformed all the other state-of-the-art methods. Furthermore, non-parametric statistical tests are used to prove the significant difference of the algorithm, namely the Friedman test, and Iman and Davenport's test. After conducting the non-parametric statistical tests, the Holm step-down procedure is used as a post-hoc procedure. The obtained statistical results prove the significant difference in between the results obtained by HEFFF and the other competitor approaches.

In the second simulation, the algorithm is employed for the text document clustering problem, where six standard text documents are used for evaluation purposes. The obtained results are compared to other non-metaheuristic and metaheuristic methods (K-mean, K-mean++, DBSCAN, Agglomerative, Spectral, KHA, HS, PSO, GA, MVO, H-PSO, H-GA, H-MVO1, H-MVO2). For evaluation purposes, the error rate, accuracy, precision, recall, F-measure, purity, and entropy are used in the text document clustering experiments. The proposed hybrid FFO demonstrates much better results over the other state-of-the-art methods and that it is, therefore, suitable for document clustering problems.

The proposed technique combines benefits from FFA and FA. The additional combination with the traditional K-means algorithm leads to a framework that delivers the best performance for text-based document clustering, as observed in the obtained results.

In future work, we plan to include other text-based document datasets in the experiments, as well as to implement other algorithms and to additionally incorporate different mechanisms for improvement and to enhance their efficiency in text document clustering.

Based on the used benchmark tests, our method obtains good results, generally outperforming the outputs obtained by other state-of-the-art techniques. Thus, we are confident that it can be adapted and tested on other NP-hard tasks, even different from of clustering, such as machine learning hyperparameter tuning, feature selection [72], wireless sensor network localization, intrusion detection within networks [73,74], energy, cloud task scheduling, path planning, portfolio optimization, etc.

Author Contributions: Conceptualization, T.B., N.B. and C.S.; methodology, N.B., C.S. and K.V.; software, T.B. and K.V.; validation, A.A.N.; formal analysis, M.Z.; investigation, T.A.R. and N.B.; resources, T.B. and C.S.; data curation, M.Z., C.S. and N.B.; writing—original draft preparation, T.B. and T.A.R.; writing—review and editing, C.S., T.A.R. and A.A.N.; visualization, T.B.; supervision, A.A.N.; project administration, M.Z. and N.B.; funding acquisition, N.B. and C.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by Ministry of Education and Science of Republic of Serbia, Grant No. III-44006. Catalin Stoean acknowledges the support of a grant of the Romanian Ministry of Education and Research, CCCDI—UEFISCDI, project number 411PED/2020, code PN-III-P2-2.1-PED-2019-2271, within PNCDI III.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Datasets used in experiments are public and they were retrieved from the following URLs: <http://archive.ics.uci.edu/ml/index.php/>, <https://www.kaggle.com/ammrabbasi/20newsgroups-300-articles>, <http://archive.ics.uci.edu/ml/index.php/>, <http://archive.ics.uci.edu/ml/index.php/>, <https://www.kaggle.com/benhamner/exploring-the-nips-2015-papers/data>, <https://www.archive.ics.uci.edu/ml> (accessed on 20 June 2021).

Acknowledgments: This research is supported by Ministry of Education and Science of Republic of Serbia, Grant No. III-44006. Catalin Stoean acknowledges the support of a grant of the Romanian Ministry of Education and Research, CCCDI—UEFISCDI, project number 411PED/2020, code PN-III-P2-2.1-PED-2019-2271, within PNCDI III.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Pan, W.T. A new evolutionary computation approach: Fruit fly optimization algorithm. In Proceedings of the 2011 Conference of Digital Technology and Innovation Management, Shenzhen, China, 26–27 November 2011; pp. 382–391.
- Pan, W.T. A new fruit fly optimization algorithm: Taking the financial distress model as an example. *Knowl. Based Syst.* **2012**, *26*, 69–74. [\[CrossRef\]](#)
- Yang, X.S. Firefly Algorithms for Multimodal Optimization. In *Stochastic Algorithms: Foundations and Applications*; Watanabe, O., Zeugmann, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 169–178.
- MacQueen, J.B. Some Methods for Classification and Analysis of MultiVariate Observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*; Cam, L.M.L., Neyman, J., Eds.; University of California Press: Berkeley, CA, USA, 1967; Volume 1, pp. 281–297.
- Xu, D.; Tian, Y. A comprehensive survey of clustering algorithms. *Ann. Data Sci.* **2015**, *2*, 165–193. [\[CrossRef\]](#)
- Sun, W.; Tang, M.; Zhang, L.; Huo, Z.; Shu, L. A Survey of Using Swarm Intelligence Algorithms in IoT. *Sensors* **2020**, *20*, 1420. [\[CrossRef\]](#) [\[PubMed\]](#)
- Park, H.S.; Jun, C.H. A simple and fast algorithm for K-medoids clustering. *Expert Syst. Appl.* **2009**, *36*, 3336–3341. [\[CrossRef\]](#)
- Kaufman, L.; Rousseeuw, P.J. *Finding Groups in Data: An Introduction to Cluster Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2009; Volume 344.
- Ng, R.T.; Han, J. CLARANS: A method for clustering objects for spatial data mining. *IEEE Trans. Knowl. Data Eng.* **2002**, *14*, 1003–1016. [\[CrossRef\]](#)
- Zhang, T.; Ramakrishnan, R.; Livny, M. BIRCH: An efficient data clustering method for very large databases. *ACM Sigmod Rec.* **1996**, *25*, 103–114. [\[CrossRef\]](#)
- Guha, S.; Rastogi, R.; Shim, K. CURE: An efficient clustering algorithm for large databases. *ACM Sigmod Rec.* **1998**, *27*, 73–84. [\[CrossRef\]](#)
- Guha, S.; Rastogi, R.; Shim, K. ROCK: A robust clustering algorithm for categorical attributes. *Inf. Syst.* **2000**, *25*, 345–366. [\[CrossRef\]](#)
- Karypis, G.; Han, E.H.; Kumar, V. Chameleon: Hierarchical clustering using dynamic modeling. *Computer* **1999**, *32*, 68–75. [\[CrossRef\]](#)
- Bezdek, J.C.; Ehrlich, R.; Full, W. FCM: The fuzzy c-means clustering algorithm. *Comput. Geosci.* **1984**, *10*, 191–203. [\[CrossRef\]](#)
- Dave, R.N.; Bhaswan, K. Adaptive fuzzy c-shells clustering and detection of ellipses. *IEEE Trans. Neural Netw.* **1992**, *3*, 643–662. [\[CrossRef\]](#)
- Yager, R.R.; Filev, D.P. Approximate clustering via the mountain method. *IEEE Trans. Syst. Man Cybern.* **1994**, *24*, 1279–1284. [\[CrossRef\]](#)
- Xu, X.; Ester, M.; Kriegel, H.P.; Sander, J. A distribution-based clustering algorithm for mining in large spatial databases. In Proceedings of the 14th International Conference on Data Engineering, Orlando, FL, USA, 23–27 February 1998; pp. 324–331.
- Rasmussen, C.E. The infinite Gaussian mixture model. *NIPS* **1999**, *12*, 554–560.
- Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96, Portland, OR, USA, 2–4 August 1996; AAAI Press: Palo Alto, CA, USA, 1996; p. 226–231.
- Ankerst, M.; Breunig, M.M.; Kriegel, H.P.; Sander, J. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod Rec.* **1999**, *28*, 49–60. [\[CrossRef\]](#)
- Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 603–619. [\[CrossRef\]](#)
- Sharan, R.; Shamir, R. CLICK: A Clustering Algorithm with Applications to Gene Expression Analysis. In Proceedings International Conference on Intelligent Systems for Molecular Biology, ISMB, San Diego, CA, USA, 19–23 August 2000; International Conference on Intelligent Systems for Molecular Biology; AAAI Press: Palo Alto, CA, USA, 2000; Volume 8, pp. 307–316.
- Gupta, M.K.; Chandra, P. A comparative study of clustering algorithms. In Proceedings of the 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 13–15 March 2019; pp. 801–805.
- Wang, W.; Yang, J.; Muntz, R. STING: A statistical information grid approach to spatial data mining. *VLDB*, **1997**, *97*, 186–195.
- Agrawal, R.; Gehrke, J.; Gunopulos, D.; Raghavan, P. Automatic subspace clustering of high dimensional data for data mining applications. In Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, Seattle, WA, USA, 1–4 June 1998; pp. 94–105.
- Barbará, D.; Chen, P. Using the fractal dimension to cluster datasets. In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, USA, 20–23 August 2000; pp. 260–264.
- Fisher, D.H. Knowledge acquisition via incremental conceptual clustering. *Mach. Learn.* **1987**, *2*, 139–172. [\[CrossRef\]](#)
- Hrosik, R.C.; Tuba, E.; Dolicanin, E.; Jovanovic, R.; Tuba, M. Brain Image Segmentation Based on Firefly Algorithm Combined with K-means Clustering. *Stud. Inf. Control* **2019**, *28*, 167–176. [\[CrossRef\]](#)
- Tuba, E.; Strumberger, I.; Bezdan, T.; Bacanin, N.; Tuba, M. Classification and feature selection method for medical datasets by brain storm optimization algorithm and support vector machine. *Procedia Comput. Sci.* **2019**, *162*, 307–315. [\[CrossRef\]](#)

30. Tuba, E.; Strumberger, I.; Bacanin, N.; Bezdan, T.; Tuba, M. Image Clustering by Generative Adversarial Optimization and Advanced Clustering Criteria. In Proceedings of the International Conference on Swarm Intelligence, Belgrade, Serbia, 14–20 July 2020; Springer: Berlin, Germany, 2020; pp. 465–475.
31. Brajevic, I.; Tuba, M.; Bacanin, N. Multilevel image thresholding selection based on the cuckoo search algorithm. In Proceedings of the 5th International Conference on Visualization, Imaging and Simulation (VIS'12), Sliema, Malta, 7–9 September 2012; pp. 217–222.
32. Lichtblau, D.; Stoean, C. Cancer diagnosis through a tandem of classifiers for digitized histopathological slides. *PLoS ONE* **2019**, *14*, e.0209274. [\[CrossRef\]](#)
33. Bezdan, T.; Cvetnic, D.; Gajic, L.; Zivkovic, M.; Strumberger, I.; Bacanin, N. Feature Selection by Firefly Algorithm with Improved Initialization Strategy. In Proceedings of the 7th Conference on the Engineering of Computer Based Systems, Novi Sad, Serbia, 26–27 May 2021; Association for Computing Machinery: New York, NY, USA, 2021; ECBS 2021. [\[CrossRef\]](#)
34. Rana, S.; Jasola, S.; Kumar, R. A review on particle swarm optimization algorithms and their applications to data clustering. *Artif. Intell. Rev.* **2011**, *35*, 211–222. [\[CrossRef\]](#)
35. Strumberger, I.; Tuba, E.; Bacanin, N.; Zivkovic, M.; Beko, M.; Tuba, M. Designing Convolutional Neural Network Architecture by the Firefly Algorithm. In Proceedings of the 2019 International Young Engineers Forum (YEF-ECE), Costa da Caparica, Portugal, 10 May 2019; pp. 59–65. [\[CrossRef\]](#)
36. Bezdan, T.; Tuba, E.; Strumberger, I.; Bacanin, N.; Tuba, M. Automatically Designing Convolutional Neural Network Architecture with Artificial Flora Algorithm. In *ICT Systems and Sustainability*; Tuba, M., Akashe, S., Joshi, A., Eds.; Springer: Singapore, 2020; pp. 371–378.
37. Bacanin, N.; Bezdan, T.; Tuba, E.; Strumberger, I.; Tuba, M. Monarch Butterfly Optimization Based Convolutional Neural Network Design. *Mathematics* **2020**, *8*, 936. [\[CrossRef\]](#)
38. Bacanin, N.; Bezdan, T.; Tuba, E.; Strumberger, I.; Tuba, M. Optimizing Convolutional Neural Network Hyperparameters by Enhanced Swarm Intelligence Metaheuristics. *Algorithms* **2020**, *13*, 67. [\[CrossRef\]](#)
39. Stoean, R. Analysis on the potential of an EA-surrogate modelling tandem for deep learning parametrization: An example for cancer classification from medical images. *Neural Comput. Appl.* **2018**, *32*, 313–322. [\[CrossRef\]](#)
40. 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* **2019**, *19*, 2515. [\[CrossRef\]](#)
41. Zivkovic, M.; Bacanin, N.; Tuba, E.; Strumberger, I.; Bezdan, T.; Tuba, M. Wireless Sensor Networks Life Time Optimization Based on the Improved Firefly Algorithm. In Proceedings of the 2020 International Wireless Communications and Mobile Computing (IWCMC), Limassol, Cyprus, 15–19 June 2020; pp. 1176–1181.
42. Bacanin, N.; Bezdan, T.; Tuba, E.; Strumberger, I.; Tuba, M.; Zivkovic, M. Task Scheduling in Cloud Computing Environment by Grey Wolf Optimizer. In Proceedings of the 2019 27th Telecommunications Forum TELFOR, Belgrade, Serbia, 26–27 November 2019; pp. 1–4.
43. Bacanin, N.; Tuba, E.; Bezdan, T.; Strumberger, I.; Tuba, M. Artificial flora optimization algorithm for task scheduling in cloud computing environment. In Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning, Manchester, UK, 14–16 November 2019; Springer: Berlin, Germany, 2019; pp. 437–445.
44. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [\[CrossRef\]](#)
45. Bezdan, T.; Zivkovic, M.; Antonijevic, M.; Zivkovic, T.; Bacanin, N. Enhanced Flower Pollination Algorithm for Task Scheduling in Cloud Computing Environment. In *Machine Learning for Predictive Analysis*; Springer: Berlin, Germany, 2020; pp. 163–171.
46. Bezdan, T.; Zivkovic, M.; Tuba, E.; Strumberger, I.; Bacanin, N.; Tuba, M. Multi-objective Task Scheduling in Cloud Computing Environment by Hybridized Bat Algorithm. In *Intelligent and Fuzzy Techniques: Smart and Innovative Solutions*; Kahraman, C., Cevik Onar, S., Oztaysi, B., Sari, I.U., Cebi, S., Tolga, A.C., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 718–725.
47. Strumberger, I.; Bacanin, N.; Tuba, M.; Tuba, E. Resource scheduling in cloud computing based on a hybridized whale optimization algorithm. *Appl. Sci.* **2019**, *9*, 4893. [\[CrossRef\]](#)
48. Bitam, S.; Mellouk, A.; Zeadally, S. Bio-Inspired Routing Algorithms Survey for Vehicular Ad Hoc Networks. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 843–867. [\[CrossRef\]](#)
49. Marinakis, Y.; Iordanidou, G.R.; Marinaki, M. Particle Swarm Optimization for the Vehicle Routing Problem with Stochastic Demands. *Appl. Soft Comput.* **2013**, *13*, 1693–1704. [\[CrossRef\]](#)
50. Tuba, E.; Strumberger, I.; Zivkovic, D.; Bacanin, N.; Tuba, M. Mobile Robot Path Planning by Improved Brain Storm Optimization Algorithm. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8. [\[CrossRef\]](#)
51. Lodi, A.; Martello, S.; Vigo, D. Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. *INFORMS J. Comput.* **1999**, *11*, 345–357. [\[CrossRef\]](#)
52. Madni, S.H.H.; Latiff, M.S.A.; Coulibaly, Y.; Abdulhamid, S.M. An appraisal of meta-heuristic resource allocation techniques for IaaS cloud. *Indian J. Sci. Technol.* **2016**, *9*, 1–14. [\[CrossRef\]](#)

53. Bezdan, T.; Zivkovic, M.; Tuba, E.; Strumberger, I.; Bacanin, N.; Tuba, M. Glioma Brain Tumor Grade Classification from MRI Using Convolutional Neural Networks Designed by Modified FA. In Proceedings of the International Conference on Intelligent and Fuzzy Systems, Istanbul, Turkey, 21–23 July 2020; Springer: Berlin, Germany, 2020; pp. 955–963.
54. Bacanin, N.; Tuba, E.; Bezdan, T.; Strumberger, I.; Jovanovic, R.; Tuba, M. Dropout Probability Estimation in Convolutional Neural Networks by the Enhanced Bat Algorithm. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–7. [\[CrossRef\]](#)
55. Gajic, L.; Cvetnic, D.; Zivkovic, M.; Bezdan, T.; Bacanin, N.; Milosevic, S. Multi-layer Perceptron Training Using Hybridized Bat Algorithm. In *Computational Vision and Bio-Inspired Computing*; Smys, S., Tavares, J.M.R.S., Bestak, R., Shi, F., Eds.; Springer: Singapore, 2021; pp. 689–705.
56. Milosevic, S.; Bezdan, T.; Zivkovic, M.; Bacanin, N.; Strumberger, I.; Tuba, M. Feed-Forward Neural Network Training by Hybrid Bat Algorithm. In Proceedings of the Modelling and Development of Intelligent Systems: 7th International Conference, MDIS 2020, Sibiu, Romania, 22–24 October 2020; Revised Selected Papers 7; Springer International Publishing: Basel, Switzerland, 2021; pp. 52–66.
57. Bacanin, N.; Bezdan, T.; Venkatachalam, K.; Al-Turjman, F. Optimized convolutional neural network by firefly algorithm for magnetic resonance image classification of glioma brain tumor grade. *J. Real Time Image Process.* **2021**. [\[CrossRef\]](#)
58. Griffis, S.E.; Bell, J.E.; Closs, D.J. Metaheuristics in logistics and supply chain management. *J. Bus. Logist.* **2012**, *33*, 90–106. [\[CrossRef\]](#)
59. Xiao, W.; Yang, Y.; Xing, H.; Meng, X. Clustering Algorithm Based on Fruit Fly Optimization. In Proceedings of the International Conference on Rough Sets and Knowledge Technology, Tianjin, China, 20–23 November 2015; Springer: Cham, Switzerland, 2015; pp. 408–419. [\[CrossRef\]](#)
60. Chen, H.N.; He, B.; Yan, L.; Li, J.; Ji, W. A text clustering method based on two-dimensional OTSU and PSO algorithm. In Proceedings of the 2009 International Symposium on Computer Network and Multimedia Technology, Wuhan, China, 18–20 January 2009; pp. 1–4.
61. Purushothaman, R.; Rajagopalan, S.; Dhandapani, G. Hybridizing Gray Wolf Optimization (GWO) with Grasshopper Optimization Algorithm (GOA) for text feature selection and clustering. *Appl. Soft Comput.* **2020**, *96*, 106651. [\[CrossRef\]](#)
62. Nanda, S.J.; Panda, G. A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm Evol. Comput.* **2014**, *16*, 1–18. [\[CrossRef\]](#)
63. Abualigah, L.M.; Khader, A.T.; Al-Betar, M.A.; Awadallah, M.A. A krill herd algorithm for efficient text documents clustering. In Proceedings of the 2016 IEEE symposium on computer applications & industrial electronics (ISCAIE), Penang, Malaysia, 30–31 May 2016; pp. 67–72.
64. 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 Comput. Appl.* **2020**, *32*, 17703–17729. [\[CrossRef\]](#)
65. Wang, H.; Zhou, X.; Sun, H.; Yu, X.; Zhao, J.; Zhang, H.; Cui, L. Firefly algorithm with adaptive control parameters. *Soft Comput.* **2017**, *3*, 5091–5102. [\[CrossRef\]](#)
66. Price, K.; Awad, N.; Ali, M.; Suganthan, P. Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization. In *Technical Report*; Nanyang Technological University: Singapore, 2018.
67. Muthusamy, H.; Ravindran, S.; Yaacob, S.; Polat, K. An improved elephant herding optimization using sine–cosine mechanism and opposition based learning for global optimization problems. *Expert Syst. Appl.* **2021**, *172*, 114607. [\[CrossRef\]](#)
68. Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. D* **1937**, *32*, 675–701. [\[CrossRef\]](#)
69. Friedman, M. A comparison of alternative tests of significance for the problem of m rankings. *Ann. Math. Stat.* **1940**, *11*, 86–92. [\[CrossRef\]](#)
70. Sheskin, D.J. *Handbook of Parametric and Nonparametric Statistical Procedures*; Chapman and Hall/CRC: Ballyclare, UK, 2020.
71. Iman, R.L.; Davenport, J.M. Approximations of the critical region of the fbietkan statistic. *Commun. Stat. Theory Methods* **1980**, *9*, 571–595. [\[CrossRef\]](#)
72. Stoean, C. In Search of the Optimal Set of Indicators when Classifying Histopathological Images. In Proceedings of the 2016 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), Timisoara, Romania, 24–27 September 2016; pp. 449–455. [\[CrossRef\]](#)
73. Kumar, M.; Mukherjee, P.; Verma, K.; Verma, S.; Rawat, D.B. Improved Deep Convolutional Neural Network based Malicious Node Detection and Energy-Efficient Data Transmission in Wireless Sensor Networks. *IEEE Trans. Netw. Sci. Eng.* **2021**, *1*. [\[CrossRef\]](#)
74. Panigrahi, R.; Borah, S.; Bhoi, A.K.; Ijaz, M.F.; Pramanik, M.; Kumar, Y.; Jhaveri, R.H. A Consolidated Decision Tree-Based Intrusion Detection System for Binary and Multiclass Imbalanced Datasets. *Mathematics* **2021**, *9*, 751. [\[CrossRef\]](#)