Computers, Materials & Continua CMC, vol.61, no.3, pp.1045-1074, 2019

Genetic-Frog-Leaping Algorithm for Text Document Clustering

Lubna Alhenak[1] and Manar Hosny[1,*]

Abstract: In recent years, the volume of information in digital form has increased tremendously owing to the increased popularity of the World Wide Web. As a result, the use of techniques for extracting useful information from large collections of data, and particularly documents, has become more necessary and challenging. Text clustering is such a technique; it consists in dividing a set of text documents into clusters (groups), so that documents within the same cluster are closely related, whereas documents in different clusters are as different as possible. Clustering depends on measuring the content (i.e., words) of a document in terms of relevance. Nevertheless, as documents usually contain a large number of words, some of them may be irrelevant to the topic under consideration or redundant. This can confuse and complicate the clustering process and make it less accurate. Accordingly, feature selection methods have been employed to reduce data dimensionality by selecting the most relevant features. In this study, we developed a text document clustering optimization model using a novel genetic frog-leaping algorithm that efficiently clusters text documents based on selected features. The proposed approach is based on two metaheuristic algorithms: a genetic algorithm (GA) and a shuffled frog-leaping algorithm (SFLA). The GA performs feature selection, and the SFLA performs clustering. To evaluate its effectiveness, the proposed approach was tested on a well-known text document dataset: the "20Newsgroup" dataset from the University of California Irvine Machine Learning Repository. Overall, after multiple experiments were compared and analyzed, it was demonstrated that using the proposed algorithm on the 20Newsgroup dataset greatly facilitated text document clustering, compared with classical K-means clustering. Nevertheless, this improvement requires longer computational time.

Keywords: Text documents clustering, meta-heuristic algorithms, shuffled frog-leaping algorithm, genetic algorithm, feature selection.

1 Introduction

In the big data era, massive amounts of information are encountered. This information should be analyzed and managed by classifying or grouping it into related categories or clusters so that it may be useful for further processing or knowledge discovery [Xu and WunschII (2005)]. Clustering can be defined as the process of forming groups of objects, so that the objects belonging to the same group are as similar as possible to one another

[1] Computer Science Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia.

[*] Corresponding Author: Lubna Alhenaki. Email: lubna.henaki@gmail.com.

and as different as possible from the objects in other groups [Aggarwal and Reddy (2014)]. In fact, clustering is a widely used technique for solving a variety of difficult problems in various common application areas, such as biological and medical data analysis, marketing, information retrieval, web browsing, and social media data analysis [Xu and WunschII (2005)]. A considerable amount of information, particularly in web resources, is often presented as text documents. As a result, the use of clustering techniques to classify these massive amounts of text into meaningful groups for extracting useful information has become a challenging task that requires efficient and robust methods [Kaur and Rohil (2015)].

The main objective of text document clustering is the automatic organization of a collection of text documents into groups, so that documents in one group are more similar in topic/content to one another than to those in other groups, depending on the similarity of their content [Xu and WunschII (2005)]. Document content is usually determined by the set of words in the document. However, using all words in a document to measure the relatedness between documents is not practical, because some of these words may be irrelevant to the topic under consideration or redundant. Therefore, data clustering techniques often require a feature selection method to select the most relevant words (i.e., terms or keywords) to be considered in the clustering so that data dimensionality may be reduced; this significantly aids in saving time and computing resources, and renders the clustering process more meaningful [Santra and Christy (2012); Dhillon, Kogan and Nicholas (2004)].

Developing a document clustering framework for a large number of text datasets is primarily motivated by the above need. In this study, we present a novel genetic-frog-leaping algorithm (GA-SFLA) for text document clustering. Two meta-heuristic algorithms are proposed for the clustering task: a genetic algorithm (GA) [Holland (1975)] performs feature selection, and a shuffled frog-leaping algorithm (SFLA) [Eusuff, Lansey and Pasha (2006)] performs clustering. We use meta-heuristic algorithms to scale down the large search space for each method owing to the complexity of the problem [Amiri, Fattah and Maroosi (2009)]. The choice of an SFLA for text document clustering is based on the success of this method in clustering problems in general (e.g., [Fang and Yu (2011); Amiri, Fattah and Maroosi (2009) ; Kalashami and Chabok (2016); Bhaduri and Bhaduri (2009)]), whereas the GA has been successful in feature selection in different contexts (e.g., [Santra and Christy (2012); Al-Jadir, Wong and Fung et al. (2017); Liu, Kang and Yu et al. (2005); Abualigah, Khader and Al-Betar (2016) Hong, Lee and Han (2015)]). However, to the best of our knowledge, an SFLA has not been used for text document clustering. Furthermore, combining GA and SFLA for feature selection and text document clustering, respectively, has not been previously attempted.

The main contribution of this study is the development of a new framework for optimizing both feature selection and text document clustering. Both optimization stages are combined to obtain the best overall solution. GA-SFLA generates groups of highly correlated text documents based on the cosine document similarity measure, as will be explained later. In addition, after multiple experiments on the well-known 20Newsgroup dataset, it was demonstrated that the proposed algorithm outperforms the classical *K*-

means clustering. Thus, using GA-SFLA on the text document dataset can greatly facilitate text document clustering.

The paper is organized as follows: Section 2 provides a background of basic related subjects. Section 3 provides a literature review. Section 4 introduces the design methodology, and Section 5 introduces the data, experimental setup, and implementation. In Section 6, we discuss the results. Finally, Section 7 concludes the paper.

2 Background

This section briefly explains the main similarity measures that are used for calculating the proximity between objects, some measures that are used to evaluate clustering quality, and an overview of the SFLA that is used for clustering in this study. The framework of the GA is well known and will not be presented in this section. For more details about the GA, the reader is referred to Goldberg [Goldberg (1989)].

2.1 Clustering proximity measures

As previously mentioned, clustering is the grouping of similar objects. For this reason, a certain measure is required to assess the degree of closeness and separation among the data objects. In fact, the quality of several algorithms depends on selecting a suitable similarity/distance measure that is linked to the data [Torres, Basnet, Sung et al. (2009); Huang (2008)]. However, there is no measure that is universally adopted for all clustering problems. For example, to measure distance, the Minkowski, Manhattan, or Euclidean distance can be used, whereas the cosine similarity is a widely used similarity measure. In this study, we adopted the Cosine similarity measure. It is defined by the dot product of the angle between two vectors, as shown by the following equation [Huang (2008); Jagatheeshkumar and Brunda (2017); Iglesias and Kastner (2013)]:

$$\cos(\vec{q},\vec{d}) = \frac{\sum_{i=1}^{|V|} q_i \cdot d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\sqrt{\sum_{i=1}^{|V|} d_i^2}} \tag{1}$$

where q_i and d_i are the term frequency-inverse document frequency (TF-IDF) weights of term i in the documents q and d, respectively, and $|V|$ is the number of features in the document [Salton and Buckley (1988)]. Each document is represented as a vector, as will be explained in more detail in Section 4.3.

2.2 Clustering validity assessment measures

Evaluating the clustering results is highly important for determining clustering quality. Clustering validity measures are generally classified into three types: internal, relative, and external [Halkidi, Batistakis and Vazirgiannis (2001)]. External measures are applicable when the ground truth data are available, whereas internal measures (indexes) are used when there is no prior knowledge about the data. Finally, a relative index is based on a comparison of the resulting clusters when the same algorithm is run repetitively with modified input values.

We focus here on internal measures that are evaluated according to the degree of relatedness of the objects in a cluster as well as the distance between clusters. That is, the internal index is based on maximizing intra-cluster similarity and minimizing inter-

cluster similarity, as mentioned in Section 2.1. For this purpose, the sum of squared error (SSE) is the most widely used internal measure. It is calculated by the following equation [Han, Kamber and Pei (2006); Davies and Bouldin (1979)]:

$$SSE\ (C_k) = \frac{1}{|C_k|}\sum_{\forall x \in C_k} Sim(x, r_k)^2, \quad (2)$$

where C_k is a cluster, x is an object in the cluster, r_k is the centroid (mean of all objects) or the medoid (a randomly selected object) of cluster k, and Sim is some similarity measure that is used to assess the similarity between two objects (e.g., cosine similarity), as explained in Section 2.1. Then, to obtain an indicator of the overall cluster quality, we calculate the ratio between the similarity within a cluster (WC) and similarity between clusters (BC), as shown in the following equation:

$$Q = \frac{WC}{BC}, \quad (3)$$

where WC is calculated by

$$WC = \sum_{k=1}^{k} SSE(C_k). \quad (4)$$

Here, SSE is calculated by Eq. (2) and BC is calculated by

$$BC = \sum_{1 \le j < k \le K} Sim\left(r_j, r_k\right)^2, \quad (5)$$

where r_j, r_k are the centroids (or medoids) of the clusters.

A higher value of Q implies better clustering quality because Q indicates the maximum similarity value within the cluster (WC), and the minimum similarity value between clusters (BC).

2.3 Shuffled frog-leaping algorithm

The SFLA is a robust meta-heuristic optimization method. It was first used in Eusuff et al. [Eusuff and Lansey (2003)]. Its principle is based on a group of frogs jumping on a number of stones in a swamp to search for the stone with the maximum amount of available food. Unlike the traditional GA that does not include local search, the SFLA performs both local and global search [Rao and Savsani (2012)]. The SFLA has been successfully applied to solve different optimization problems with reasonable processing time and cost. For instance, it has been used for determining optimal water resource distribution [Eusuff and Lansey (2003)], for data clustering [Amiri, Fattah and Maroosi (2009)], in the line sequencing problem [Ramya and Chandrasekaran (2013)], for job-shop scheduling arrangement, and in the traveling salesman problem [Wang and Di (2010)].

The details of the SFLA are as follows [Binitha and Sathya (2012); Elbeltagi, Hegazy and Grierson (2005)]. First, a random population of P solutions (frogs) is generated. Then, for each individual frog, we calculate its performance index, that is, the fitness function (FF). Subsequently, the frogs are ranked in decreasing order according to their fitness value and stored in an array X, so that a frog i is represented as $x_i\ (x_{i1}, x_{i2}, \ldots., x)$; the frog with the best performance value corresponds to i=1 (the first position of the array X) and is called the global best (P_x). Then, we partition X. That is, we separate the entire population into a number of memeplexes, each containing a number of frogs, which are denoted by m and n (i.e., P=m×n), and Y(i=1, …, m) is the array of the memeplexes. This procedure can by described as follows [Eusuff and Lansey (2003); Karakoyun and Babalik (2015)]:

$$Y_k = [(x_i) \mid x_i = X(k + m * (i - 1)), i = 1, \dots, n], k = 1, \dots, m \quad (6)$$

In this step, the first frog goes to the first memeplex, the second frog goes to the second memeplex, frog m goes to the m memeplex, and frog $m + 1$ goes back to the first memeplex.

After this stage, the local search of the SFLA begins by applying a memetic evolution process to each memeplex. However, the algorithm returns to the global search for shuffling after the memeplexes have evolved. For each memeplex, we generate a submemeplex to determine the best and the worst frogs depending on the assigned weights. Thus, the performance of the frogs in the current memeplex is an important factor in the selection strategy.

To form the submemeplex array, q distinct frogs are randomly selected from n frogs in each memeplex. Subsequently, the submemeplex is sorted in decreasing order of performance $(i_q = 1, \dots, q)$. Thus, in the submemeplex, the best frog's position is $(i_q = 1)$ and is denoted by PB, whereas the worst frog's position is $(i_q = q)$ and is denoted by Pw. Fig. 1 shows the concept of a submemeplex.

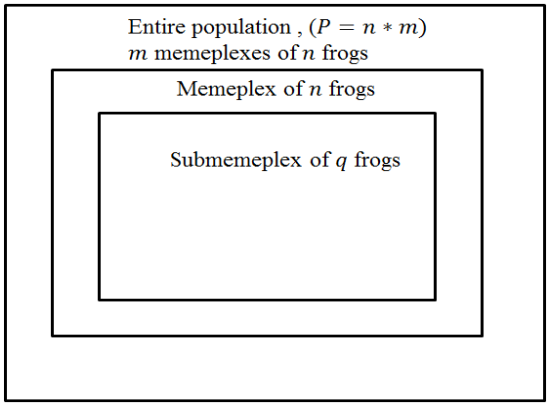

Figure 1: Submemeplex overview

To perform local search in each submemeplex, we update the worst frog by combining it through a crossover operation with the best frog (PB) in the submemeplex. This process will generate a new position in the search space (a new frog). After calculating the fitness value of the new frog, we have the following sequence of conditions:

1. If the new frog is better than the worst frog (Pw) in the submemeplex, it will replace the worst frog.
2. If Step 1 is not true, we replace PB with the global best P_x in the crossover operation and calculate the fitness value of the newly generated frog.
3. If the new frog is better than the worst frog (Pw), it will replace the worst frog.
4. If Step 3 is not true, we randomly generate a new frog and use it to replace Pw.

The purpose of the local search operation is to improve only the frog with the worst fitness (not all frogs). This is in contrast to the GA, which primarily concentrates on the best fitness values. If a stopping condition is satisfied, memetic evolution is terminated

for each memeplex (the end of local search). Then, we return to the global search shuffling process, which aims to generate new memeplexes after the memetic evolution loops terminate. We again arrange X in decreasing order according to the fitness value of the frogs and update the position of the globally best frog (P_x). The process then continues for a number of iterations until a certain stopping condition is satisfied [Eusuff and Lansey (2003)]. The pseudo-code of the basic SFLA is summarized in Algorithm (1).

Algorithm (1): SFLA

1. Initialize the parameters of the algorithm
2. Generate a random population
3. Evaluate the fitness of the population
4. Sort the members (frogs) according to performance
5. Initialize counter to 1
6. Separate the frogs into groups (memeplexes)
7. Memetic evolution for each memeplex

a. Generate a sub-memeplex from the current memeplex
b. Find the worst frog's position
c. Update the worst frog's position
d. Sort the memeplex by performance
e. If sub-iterations finish go to next memeplex
f. If evolution is performed for all memeplexes go to the next step

8. Shuffle the memeplexes and sort the population
9. Update global best
10. Increase counter by 1
11. If counter is less than max iteration number then go to Step 6

3 Related work

This section surveys and discusses related work in the domain of clustering but is not meant to be exhaustive. We categorize related work into two sections: the first is a review of clustering techniques that used the SFLA, whereas the second is a review of research related to feature selection in document clustering.

3.1 Shuffled frog-leaping algorithm in clustering techniques

Nature-inspired meta-heuristic algorithms are powerful and widely used. The SFLA is a type of a population-based evolutionary meta-heuristic algorithm that has been successfully applied in several clustering techniques. For instance, in Bhaduri et al. [Bhaduri and Bhaduri (2009)], two nature-inspired memetic meta-heuristic algorithms, namely, the SFLA and clonal selection SFLA (CSSFLA), were applied to image segmentation using the clustering method. These algorithms were used to locate the optimal image clusters and to compare the results for image segmentation. Experiments demonstrated that the CSSFLA greatly outperformed the SFLA.

In Kalashami et al. [Kalashami and Chabok (2016)], the SFLA was improved by replacing the random behavior of the algorithm with chaotic behavior and combination operators in the local search. An experiment was conducted using four real datasets: Iris, Wine, Glass, and Cancer from the University of California Irvine (UCI) Machine Learning Repository ("UC Irvine Machine Learning Repository," n.d.). The effectiveness of all algorithms in terms of error rate was evaluated, and it was demonstrated that, compared with the GA as well as the K-means and the particle swarm optimization (PSO) algorithms, this method is superior and can be efficiently used in clustering problems.

In Amiri et al. [Amiri, Fattah and Maroosi (2009)], an application of the SFLA and the K-means algorithm (SFLK-means) was proposed for clustering. The proposed algorithm was tested on four artificial datasets and five real-life datasets (Vowel, Iris, Crude Oil, Wine, and Thyroid diseases data). Moreover, SFLK-means was compared with the ant colony optimization algorithm, a simulated annealing (SA) approach, a genetic K-means algorithm, and Tabu search. The results demonstrated that the proposed algorithm performed better than the other approaches.

In Fang et al. [Fang and Yu (2011)], the K-means algorithm was used for web documents clustering. In this algorithm, it is difficult to select the optimal number of clusters and the initial centroids; therefore, the SFLA was used for the selection of the K value. The proposed algorithm was tested on XML documents that can be expressed by, for instance, their title, keyword list, or abstract. The results demonstrated that using the K-means clustering algorithm with the SFLA improved the clustering performance in terms of accuracy.

In Zhang et al. [Zhang, Liu, Liang et al. (2016)], a K-means algorithm with multiple swarm intelligence based on SA, the SFLA, and PSO was proposed. The proposed algorithm combines the local search mechanism of the SFLA with the global optimization capability of PSO and adjusts the parameters by an optimization method using SA. An experiment was conducted using the Wine dataset from the UCI database. In addition, this experiment compared the proposed algorithm with the SFLA and the K-means algorithm with shuffled frog leaping. It was demonstrated that the proposed searching strategy achieved the best performance.

A recent study, Karakoyun et al. [Karakoyun and Babalik (2015)] used the SFLA for partitional data clustering. The test dataset consisted of 12 benchmark problems taken from the UCI Machine Learning Repository. Several meta-heuristic algorithms, such as the artificial bee colony algorithm and PSO, as well as nine other classification algorithms were compared with the SFLA. It was demonstrated that the SFLA was the most effective.

3.2 Feature selection in document clustering

As text documents are high-dimensional structures, feature selection is critical for selecting a subset of important features for clustering over the entire data set. Feature selection in document clustering has been addressed in numerous studies. For instance, in Liu et al. [Liu, Kang, Yu et al. (2005)], four unsupervised feature selection methods were introduced: document frequency, term contribution (TC), and term variance quality; furthermore, a new method was proposed: term variance (TV). Moreover, they used K-means as the clustering algorithm. Four different text datasets (FBIS, REI, TR45, and

TR41) were used for evaluation. The results demonstrated that feature selection based on TV and TC outperformed the other types.

In Abualigah et al. [Abualigah, Khader and Al-Betar (2016)], a feature selection technique using a GA for text clustering was proposed to find an optimal low-dimensional subset of informative features. The algorithm was tested on four common benchmark datasets, including Reuters-21578 and 20Newsgroup (David D. Lewis, n.d.) (Tom Mitchell, n.d.). In addition, the K-means clustering algorithm was used for text clustering. Furthermore, the informative features (i.e., select optimal text features) were determined using TF-IDF. It was demonstrated that the proposed feature selection achieved the best performance in most datasets.

Another related study, Hong et al. [Hong, Lee and Han (2015)] proposed a new GA called FSGA for feature selection to improve the analytical performance and speed in text mining. A set of spam mail documents from the LingSpam dataset were used for the experiments. In addition, the K-means clustering algorithm was used for text clustering, and TF-IDF was used to evaluate the selected features. The results demonstrated that the proposed FSGA is appropriate for feature selection.

In conclusion, based on the literature review above, it appears that the SFLA has not been applied to text document clustering. It has only been applied to select the best K value in K-means clustering in Fang et al. [Fang and Yu (2011)]. However, the SFLA has been applied to a closely related problem, namely, web document classification (i.e., where the class label is known) [Sun, Wang and Zhang (2008)]. Moreover, feature selection methods have been successfully applied to text document clustering in various studies [Liu, Kang and Yu et al. (2005); Hong, Lee and Han (2015)] to improve the clustering performance by removing redundant, irrelevant, or inconsistent features. Hence, motivated by these approaches, we aim to investigate a two-stage text document clustering optimization model based on GA and SFLA, as previously explained. In the next section, we will explain in detail the design of the proposed method.

4 Methodology design

This section discusses the system overview of GA-SFLA for text document clustering. Specifically, Section 4.1 provides an overview of the proposed algorithm, Section 4.2 describes the necessary preprocessing steps in detail, and finally Sections 4.3 and 4.4 describe the details of GA-SFLA for text document clustering.

4.1 System overview

The overall system overview can be seen in Fig. 2, which shows the high-level components of the GA-SFLA framework. This is divided into two stages: a dimension-reduction stage using a feature selection method (GA) and text clustering stage using the SFLA.

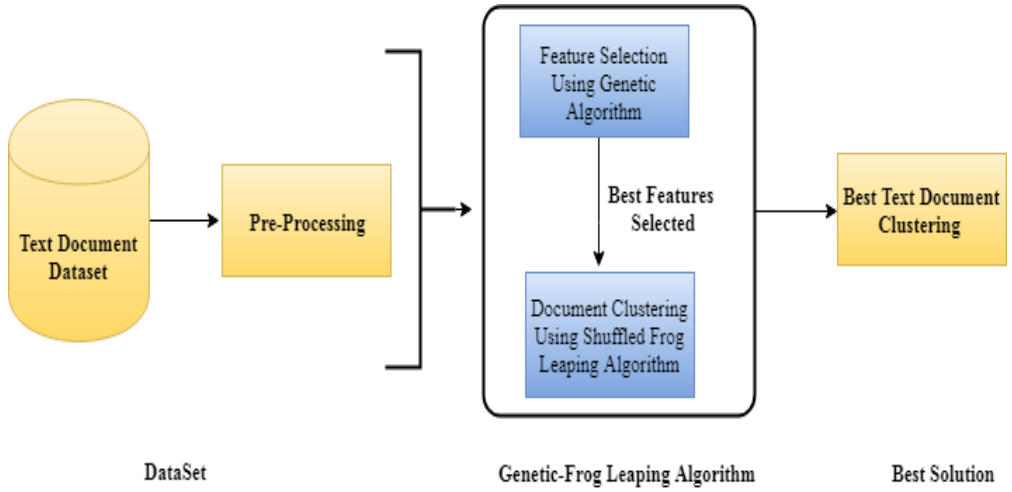

Figure 2: GA-SFLA system overview

4.2 Document preprocessing

Preprocessing is a critical step in various applications, such as natural language processing and information retrieval. Before GA-SFLA can be applied to a corpus, preliminary procedures should be performed on the text documents to clean the corpus; this will evidently improve performance and increase the effectiveness of the system.

In the proposed approach, we first perform tokenization and then remove stop words; in fact, there is no single universally acceptable list of stop words. However, some common stop-word lists are presented in Liu [Liu (2011)]. Subsequently, stemming is applied to reduce words into single terms using Porter's algorithm [Porter (1980); Willett (2006)], which is a common technique for stemming English language text documents.

4.3 Genetic algorithm for feature selection

After text document preprocessing, which decreases the dimensionality of the data, the GA feature selection phase is applied to select the most significant features from the cleaned dataset, as explained in this section. We discuss below the most important components of the GA: solution representation, FF, selection method, replacement strategy, and the crossover and mutation operators.

4.3.1 Solution representation

Chromosomes in the GA are used to represent solutions, that is, a set of selected features. Each chromosome has N genes, and each gene represents one feature (i.e., term in the document). Every chromosome contains all the features that are extracted from the documents. Particularly, features are extracted from a lexicon that is constructed after preprocessing is applied to the dataset. Moreover, each solution is represented as a binary vector with a fixed length, which is based on the number of words in the lexicon. Initially, the solution is randomly generated (i.e., when the initial population is generated). Fig. 3 shows an example of a solution for the GA feature selection step. If the

value of the gene at position j (where j is from 0 to N) is equal to l, then the j^{th} feature is selected. If the value of the gene at position j is equal to 0, the j^{th} feature is not selected. The number of selected features is set as in Liu et al. [Liu, Kang, Yu et al. (2005)]; more details are described later in Section 5.

Figure 3: Example of feature selection solution representation

4.3.2 Fitness function

The design of the FF for each solution is critical to the performance of the algorithm [Shen, Nagai and Gao (2019)]. In the proposed method, the FF depends on a weighting scheme. More specifically, the objective function, uses the TF-IDF weighting scheme that estimates the importance of a keyword not only in a particular document (locally) but rather in the entire collection of documents (globally).

We selected TF-IDF for the FF because it has been proven to be computationally efficient, as mentioned earlier in Section 3 [Abualigah, Khader and Al-Betar (2016); Hong, Lee and Han (2015); Alsaeedi, Fattah and Aloufi (2017)]. TF-IDF is the product of two statistical terms, TF and IDF, where TF is a term weighting method for calculating the number of times that a term t occurs in document d, and IDF is a term weighting method that indicates how common or rare a term is in the document. In addition, the importance increases proportionally with the number of times a word appears in the document but is decreased by the frequency of the word in the corpus (Cummins and O'Riordan (2006)]. TF-IDF is defined as follows [Salton and Buckley (1988)]:

$$W_{t,d} = tf_{t,d} \,.\, log \frac{N}{df_t}, \tag{7}$$

where $W_{t,d}$ is the weight for feature t (i.e., a word) in the document d, $tf_{t,d}$ is the TF, which represents the number of times that term t occurs in document d, and $log \frac{N}{df_t}$ is the IDF, which is obtained by dividing the total number of documents N by the number of documents containing the term df_t and taking the logarithm of that. We note that the TF-IDF value should be stored for all features in a separate dictionary that will be used as a lookup table in the calculation of the fitness of each solution. As in Hong et al. [Hong, Lee and Han (2015)], the FF for each solution is calculated by summing the weights of all selected features appearing in the chromosome. That is,

$$FF = \sum_{i=1}^{|N|} \sum_{j=1}^{|D|} W_{t_i,d_j} \cdot x_i, \tag{8}$$

where $|N|$ is the length of the chromosome (number of features in each solution), $|D|$ is the number of all documents in the corpus, W_{t_i,d_j} is the weight of term t_i, as explained in Eq. (7), d_j is the j^{th} document in the corpus, and x_i is the value of the corresponding gene in the chromosome (i.e., 1 or 0 depending on whether the corresponding feature is selected or not). A higher fitness value implies better solution quality. The FF is used to select better

solutions and produce offspring for a new generation by applying the genetic operations, that is, crossover and mutation, as will be explained next.

4.3.3 Selection method and replacement strategies

To apply the genetic operators, we propose a roulette-wheel selection method for choosing two chromosomes [Jong (1975)]. In this technique, the wheel is divided into portions corresponding to individuals, and their size depends on the fitness values. It is evident that a fitter individual has a greater portion in the pie and hence a higher chance of being selected as a parent for reproduction. This is followed by the selection process, that is, the selection of individuals who will survive (i.e., kept in the next generation after reproduction). We choose the steady-state replacement strategy, where low-fitness solutions in the population are removed and replaced by newly generated solutions. Therefore, an overlap between the two generations will be carried to the following iterations.

4.3.4 Crossover operator

The crossover operator aims to pass properties from parents to their children. There are different crossover operators. In the proposed GA, we used classical two-point crossover. In addition, crossover is usually applied with a high probability.

4.3.5 Mutation operator

The mutation operator aims to generate diversity in the genetic population. That is, it may append some new properties that are not already present in the parents. In the proposed approach, we used multi-point mutation. This operator is applied by randomly selecting more than one bits and then flipping them. The mutation operator is applied to one solution only, unlike the crossover that requires two solutions. In addition, mutation is usually applied with a low probability.

4.4 Shuffled frog-leaping algorithm for text document clustering

After the GA is applied, which generates the best solution (the best combination of the selected features), the SFLA is applied to cluster the documents based on the generated solution from the previous feature selection operation. As previously observed in Sections 1 and 3, the SFLA is chosen for document clustering because it has been applied successfully in numerous clustering techniques. We discuss below the most significant components of the proposed SFLA: solution representation, FF, memeplex evolution (crossover operator), and shuffling.

4.4.1 Solution representation

The population consists of a set of memeplexes; each memeplex has a set of frogs; each frog has a meme, and each meme has memotype(s). That is, in the proposed approach, a meme in the SFLA is used to represent a solution, whereas a memotype is used to represent a document. Furthermore, a frog is represented as a vector of decision variables (i.e., number of memotype(s) in a meme carried by a frog). Likewise, the frogs (memes) in the SFLA are equivalent to the GA chromosomes. In the proposed SFLA, the solution representation is as in Hruschka et al. [Hruschka, Campello and Freitas et al. (2009)], which is based on the medoids method (i.e., center of the cluster) that was proposed in Kaufman et al. [Kaufman and Rousseeuw (1987)].

Each solution has a length of $N + 1$, where N is the number of objects (the number of documents in the dataset), and the additional cell is for the value of the FF, which measures the goodness of the solution. To create a chromosome (solution) in the initial population, the number of clusters is specified in advance based on the corpus used, and a number of objects are randomly selected as medoids of the clusters. Then, the nearest object to each medoid is assigned to the same cluster, based on the cosine similarity, as previously described in Section 2.1. To calculate the cosine similarity, the documents are represented using the well-known vector space model, which transforms text into a vector format that contains only numbers. That is, each document is represented as a vector of term weights using TF-IDF. Thus, for each document, we have a |V|-dimensional vector space that refers to the number of the selected features (from the GA phase) that represent the documents.

Fig. 4 shows an example of a chromosome consisting of seven objects distributed into three clusters, where each cluster has one medoid represented by -1, namely index numbers 1, 2, and 4, and the remaining objects have as value the number of the cluster index to which they belong.

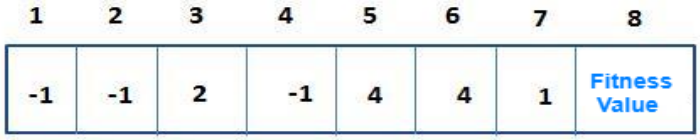

Figure 4: Example of clustering solution representation

The representation of the previous solution as a cluster diagram is shown in Fig. 5.

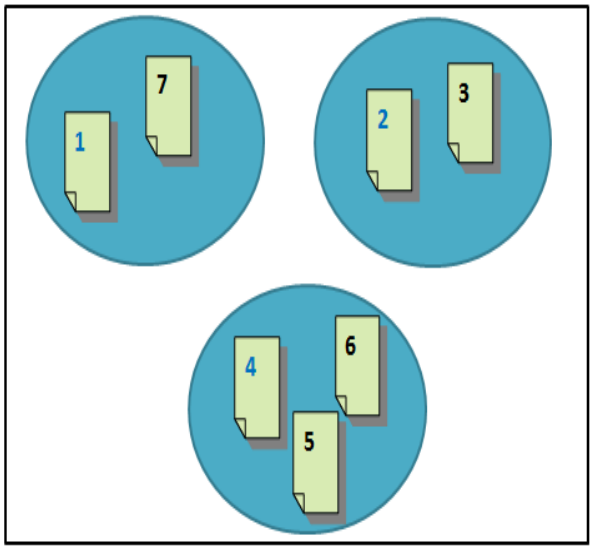

Figure 5: Cluster diagram of the previous solution in Fig. 4

After encoding the solution, we will compute the performance (fitness) value for each frog as will be described next. A number of solutions (frogs) will be generated to represent the initial population of the SFLA.

4.4.2 Fitness function

After each solution is encoded as explained in the previous section, the FF is applied to evaluate the performance of each solution. In fact, clustering quality depends on two factors: maximizing intra-cluster (within a cluster) similarity and minimizing inter-cluster (between clusters) similarity. Thus, we propose an intra-cluster and inter-cluster similarity ratio as in Han et al. [Han and Kamber (2006); Hosny, Hinti and Al-Malak (2018)] to evaluate the clustering solution, as explained in Eq. (3) in Section 2.2:

$$Q = \frac{WC}{BC} \tag{9}$$

where WC and BC are calculated by Eqs. (4) and (5) in Section 2.2, respectively.

4.4.3 Memeplex evolution (crossover operator)

Recalling the details of the SFLA in Section 2.3, we note that in each cycle of the local search, the evolution process is utilized to enhance only the worst frog (which has the worst fitness), not all frogs. This is carried out by first identifying the best and the worst frogs within each submemeplex according to their fitness. Moreover, the frog with the globally best fitness is identified. Then, an evolution process is applied using crossover. For the crossover operator, we propose here join-and-split crossover [Al-Malak and Hosny (2016)]. In this crossover, parents randomly pass down the selected medoids they carry to the child, where in our case the number of medoids is the same in each parent.

For example, if parent 1 (best frog) carries medoids 2 and 4, whereas parent 2 (worst frog) carries medoids 3 and 6, then, for example, the child will randomly carry medoids 3 and 4.

After the medoids of the child have been determined, the objects (documents) will be redistributed by assigning each object to it is nearest medoid, and then the FF for the child will be evaluated and stored in the solution. If the new child is not better than the worst frog (its parent), then we replace the best frog parent with the globally best frog, preform the crossover again between the globally best and the worst frog, and calculate the fitness function of the child again. If the new frog is still worse than the worst frog, then we randomly generate a new frog to replace the worst frog. Fig. 6 shows a crossover example.

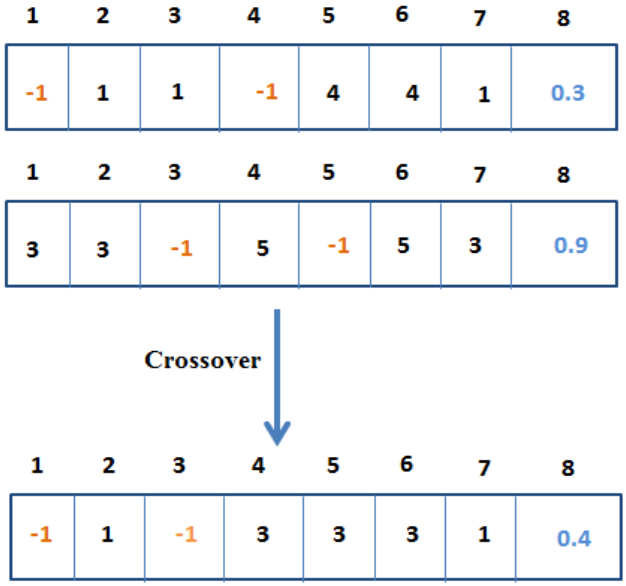

Figure 6: SFLA crossover example

4.4.4 Shuffling process

The shuffling process is applied after a number of evolution stages (local search). All frogs (solutions) of all memeplexes are collected again and arranged in descending order according to their performance value. We then repeat the previous steps until a certain convergence criterion is satisfied, and we finally determine the best clustering solution in the population.

5 Data and experimental setup

In this section, we provide details about the experimental data and setup, as well as the implementation of the proposed approach. Specifically, Section 5.1 presents the data collection method used in the experiment. Section 5.2 explains the evaluation measures used to assess the proposed approach. Section 5.3 describes the implementation of the proposed algorithm, and finally Section 5.4 explains the parameter tuning process.

5.1 Data collection

For experimental validation and performance analysis of the proposed approach, a widely used real text document dataset from UCI Machine Learning Repository is selected for

clustering: the 20Newsgroup dataset (Tom Mitchell, n.d.). The reason for choosing this dataset is that it has been widely used in related studies (as mentioned earlier in Section 3), such as Al-Jadir et al. [Al-Jadir, Wong, Fung et al. (2017); ; Abualigah, Khader and Al-Betar (2016); Karol and Mangat (2013); Patil and Atique (2013)] and contains approximately 20,000 newsgroup articles, divided across 20 different newsgroup categories, as shown in Tab. 1. Furthermore, similar categories are grouped together: for example, rec.autos, rec.motorcycles, rec.sport.baseball, and rec.sport.hockey (sports category). Thus, the dataset is suitable for clustering applications [Wang, Song, Li et al. (2018)].

Table 1: 20Newsgrroups categories

comp.graphics	rec.autos	sci.crypt
comp.os.ms-windows.misc	rec.motorcycles	sci.electronics
comp.sys.ibm.pc.hardware	rec.sport.baseball	sci.med
comp.sys.mac.hardware	rec.sport.hockey	sci.space
comp.windows.x		
misc.forsale	talk.politics.misc	talk.religion.misc
	talk.politics.guns	alt.atheism
	talk.politics.mideast	soc.religion.christian

5.2 Performance evaluation

To assess the proposed algorithm on the 20Newsgroup dataset, we quantitatively measure its performance using external (i.e., matching clustering structure to some prior knowledge) and internal (i.e., comparing different sets of clusters without prior knowledge) measures. We use the F-macro and F-micro external measures as in Al-Jadir et al. [Al-Jadir, Wong, Fung et al. (2017); Liu, Kang, Yu et al. (2005); Abualigah, Khader and Al-Betar (2016); Janani (2016); Sun, Wang and Zhang (2008)]. However, to obtain these measures, we should compute the precision and recall values. Moreover, the intra-cluster and inter-cluster similarity ratio (WC/BC) is used as an internal measure.

External measures are calculated for all documents that are grouped in a cluster, and the results are compared with ground truth data. Higher values imply a better clustering solution. However, the internal measures are different among problems.

Furthermore, we tested the proposed algorithm using different numbers of selected features and different dataset sizes. In addition, we compared the proposed algorithm with the GA-K-means algorithm, where a genetic algorithm is used in the feature selection phase and the traditional K-means is used for the clustering phase, on the same dataset. The details of these experiments are discussed in Section 6.

5.3 Implementation

The Python programming language was selected to implement the proposed approach using the Anaconda execution environment. We used an Intel® Xeon® E5-2603 CPU with 64 GB RAM running 64-bit Windows. The implementation process was divided into three steps, as mentioned in Section 4.1. In the first step, all required packages were installed, and data

preprocessing was performed. Subsequently, the GA was implemented for feature selection, and finally the SFLA was implemented for text document clustering. Regarding the preprocessing phase, the number of features before preprocessing was 173451, extracted from all 18846 documents, whereas after preprocessing, there were 70910 features. In the next section, we describe the implementation of the proposed approach in more detail.

5.4 Parameter tuning

In a meta-heuristic algorithm, reaching the best solution does not depend only on the algorithm itself but also on the calibration of the various parameters. It is also noted that no systematic parameter optimization has been recommended. In this section, we present the parameter selection for both the GA and the SFLA. For this purpose, we used a small dataset containing 200 documents (4111 features extracted) from the original dataset as a validation set. We tuned different parameter combinations several times on this small dataset to achieve the best performance. After these experiments, the best combination was selected.

5.4.1 Parameter tuning for the genetic algorithm

For the GA, four parameters should be adjusted: population size, crossover rate, mutation rate, and the number of generations. Moreover, different crossover operations should be tested. As previously mentioned, we tuned these parameters by separating the GA algorithm from the SFLA. As there is no agreement upon the parameter setting for the GA, we tested values that were proved efficient in previous optimization problems and were recommended by several authors.

In the tuning process, we tried different values of a single parameter while keeping all other parameters fixed. When we determined a value, we moved on to the next parameter. The tuning results were based on achieving the best fitness value, i.e., summing the TF-IDF weights of all selected features appearing in a chromosome. We tested the following values: The population size was set to 20, 50, and 100, and the crossover rate was set to 0.75, 0.80, and 0.90. The mutation rates were selected from, 0.01, 0.05, and 0.10. The number of generations was set to 30, 50, and 100. Three different crossover operations were evaluated: one-point, two-point, and uniform crossover. In addition, the numbers of flips in the mutation was set to 10, 20, 25, and 30. After these experimentations, the parameters of the GA were selected as follows:

- Population size: 50
- Number of generations: 100
- Crossover operator: Two-point crossover
- Crossover probability: 0.90
- Mutation operator: 20-bit flips
- Mutation probability: 0.10

5.4.2 Parameter tuning for the shuffled frog-leaping algorithm

The SFLA has several parameters that are required as input and should be adjusted: The number $\boldsymbol{P}$ of frogs, the number $\boldsymbol{m}$ of memeplexes, the number $\boldsymbol{n}$ of frogs in a memeplex, the number $\boldsymbol{q}$ of frogs in a submemeplex, the number of generations for each memeplex before shuffling (local search), and the number of shuffling iterations (global search). Again, there is no agreement upon the theoretical basis for determining these parameters. We tested the following values: $\boldsymbol{m}$ was set to 5, 10, and 20. $\boldsymbol{n}$ was set to 5, 10, and 20. $\boldsymbol{q}$ was selected from 4, 5, and 8. The number of generations for each memeplex before shuffling (local search) was set to 5, 10 and 20. Finally, three different numbers of shuffling iterations (global search) were evaluated: 10, 50, and 100. After these experimentations, the parameters of the SFLA were selected as follows:

- Number m of memeplexes: 20
- Number n of frogs in a memeplex: 20
- Number q of frogs in a submemeplex: 5
- Number of generations for each memeplex before shuffling (local search): 10
- Number of shuffling iterations (global search): 100

The next section presents the experimental results obtained after the parameters were tuned.

6 Results and discussion

To test the proposed algorithm on the 20Newsgroup dataset, we performed different experiments based on the evaluation metrics. The results were assessed by applying the internal measure (maximizing the ratio of intra-cluster to inter-cluster similarity) and the external measures (F-macro and F-micro).

For statistical comparisons, the average and the best results of five runs were obtained for each experiment. However, it is difficult to evaluate the quality of an unsupervised learning algorithm. In addition, feature selection presents an added difficulty because the resulting clusters depend on the dimensionality of the selected features, and any given feature subset may have its own clusters. Therefore, five experiments were carefully designed to evaluate the effectiveness of the proposed approach as shown in Tab. 2.

Table 2: Experiment description

Experiment Number	Experiment Name
Experiment (1)	Multiple feature selection for different dataset sizes
Experiment (2)	Multiple feature selection applied to all documents in the dataset
Experiment (3)	GA and K-means with different dataset sizes
Experiment (4)	GA and K-means tested on all documents in the dataset
Experiment (5)	SFLA without features selection tested for different dataset sizes

In Experiment (1), we performed multiple runs (five runs) of the proposed algorithm for three different numbers of selected features applied to different dataset sizes, where the corresponding datasets were randomly chosen. In Experiment (2), we performed multiple runs of the algorithm with three different numbers of selected features applied to the

entire document dataset. This is in contrast to previous studies [Al-Jadir, Wong and Fung et al. (2017); Abualigah, Khader and Al-Betar (2016); Patil and Atique (2013)], which use few document datasets for testing. Indeed, to the best of our knowledge, there is no study in which the entire dataset was considered.

In Experiment (3), to evaluate the effectiveness of GA-SFLA, we compared its results with those of a GA-K-means algorithm, where the classical K-means algorithm is used for the clustering purpose instead of the proposed SFLA, and the GA is used for features selection. In Experiment (4), to verify the general applicability of the proposed algorithm, we performed a comparison with the GA-K-means algorithm using the entire dataset. Finally, in Experiment (5), we ran the SFLA without feature selection (i.e., without the GA) on the same dataset and compared the results with those of GA-SFLA. In all experiments, we used 500, 1000, 1500, 2000, and all documents in the dataset. In what follows, we provide a more detailed explanation of each experiment.

6.1 Experiment (1)

The number of selected features is an important robustness factor in the proposed algorithm, and therefore it should be studied before the clustering results. To this end, we performed an independent test for different dataset sizes by varying the number of selected features.

Table 3: Number of features selected with respect to dataset size

Dataset size	Total number of features	Percentage of features selected	Number of features selected
500	7520	20%	1504
		30%	2256
		50%	3760
1000	11698	20%	2340
		30%	3510
		50%	5849
1500	15266	20%	3054
		30%	4580
		50%	7633
2000	19097	20%	3820
		30%	5730
		50%	9549
Entire dataset (1884)	70910	20%	14182
		30%	21273
		50%	35455

As in Liu et al. [Liu, Kang and Yu et al. (2005)], the length of the final feature set was set to approximately 20%, 30%, and 50% of the total number of extracted features. Tab. 3 shows the number of features generated using different dataset sizes (selected at random) after the preprocessing step.

After the number of selected features was determined, the GA feature selection algorithm was run and the results were compared for different dataset sizes and different numbers of selected features. Tab. 4 shows the results on the 20Newgroup dataset with different sizes: 500, 100, 1500, and 2000, and different numbers of selected features: 20%, 30%, and 50%. In this table, the results are based on the fitness value of the GA, as shown in Eq. (8) in Section 4.3.

Tab. 4 presents the average and the best fitness values in five runs. Henceforth, in all other tables, the best results in terms of internal and external measures will be highlighted in bold. In addition, the average processing time is expresses in s.

Table 4: Results of multiple runs for different dataset sizes and different numbers of features selected based on the GA fitness value

Dataset size	Number of features selected	Average fitness of five runs	Best fitness of five runs	Average processing time of five runs in s
500	20%	654.3830	766.8123	41.8219
	30%	864.1252	877.0244	28.1887
	50%	1411.3428	1433.8390	65.9011
Average		976.617	1025.8919	45.3039
1000	20%	1242.7624	1251.3871	80.1301
	30%	1792.2112	1824.0312	417.9722
	50%	2809.3619	2819.2159	2332.5194
Average		1948.1118	1964.8780	943.5405
1500	20%	1938.4375	2196.8856	704.3873
	30%	2638.2804	2706.9096	998.8634
	50%	4304.9501	4416.6799	1065.3920
Average		2960.5560	3106.8250	922.8809
2000	20%	2478.1532	2562.1108	474.0212
	30%	3547.1683	3609.2450	1070.2455
	50%	5674.4879	5741.85966	1637.5525
Average		**3899.9364**	**3971.0718**	1060.6064

From the results shown in Tab. 4, the internal evaluation measure, which is represented by the FF of the GA, exhibits high values, indicating that the GA feature selection method obtained more accurate results for the 2000-document dataset. By contrast, for the 500-, 1000-, and 1500-document sets, the GA feature selection method obtained relatively worse results in terms of the fitness values. Tab. 4 clearly indicates that the results obtained using feature selection with a large number of documents are superior to those corresponding to an insufficient number of documents. This is evident by the large gap between the averages results obtained when the number of documents is 500 and

those when the number is 2000. Nevertheless, it is clear that the processing time increased as the number of documents increased.

After presenting the results of the GA feature selection technique alone on the 20Newsgroup dataset, we present the results of GA-SFLA in Tab. 5. These results are based on the internal evaluation measure (WC/BC), where higher values correspond to a better clustering solution. Tab. 5 lists the average results of the clustering process of five runs for each collection size depending on the results obtained from Tab. 4. However, owing to the large processing time, we restrict the number of selected features in this experiment to 20% and 50% of the total number of features. Moreover, Tab. 5 provides the required clustering time in s.

Tab. 5 implies that a significant decrease in the fitness value is achieved as the dataset size decreases. Furthermore, for all dataset sizes, better results are obtained for larger numbers of selected features. Indeed, there is a significant difference between the two percentages for feature selection (20% and 50%) because the GA appears to provide a better approximation of the best and most effective text features when the number of features is increased; this affects document clustering in the following stage.

Additionally, it should be noted that with the increase in the dataset size and the number of selected features, the clustering quality improved further, leading to satisfactory results. However, the clustering time for 20% of the feature set is shorter than the clustering time for 50% of the features for all datasets. These findings indicate that, as expected, the clustering time decreases with the number of features.

Table 5: Results of multiple runs of the GA-SFLA for different dataset sizes based on the intra-cluster and inter-cluster similarity ratios

Dataset size	Number of features selected	Average fitness of five runs	Best fitness of five runs	Average processing time of five Runs in s
500	20%	110.3577	217.1657	1416.3938
	50%	117.5687	298.5155	1893.5787
Average		113.9632	257.8406	1654.9863
	20%	298.0379	477.87730	9259.6344
1000	50%	341.3758	1030.4602	10059.2930
Average		319.7068	754.16875	9659.4637
1500	20%	291.5403	455.4153	9407.1933
	50%	519.7878	1605.8567	10871.3446
Average		405.6640	1030.6360	10139.2689
	20%	588.6824	1016.2645	13848.6570
2000	50%	599.7699	1271.0224	18845.2108
Average		**594.2261**	**1143.6434**	16346.9339

6.2 Experiment (2)

The purpose of Experiment (2) was to test the proposed algorithm on the entire dataset with different numbers of features. As previously mentioned, to the best of our knowledge, no feature selection algorithm has been tested on the entire 20Newsgroup

dataset. Therefore, we were motivated to conduct this experiment, which we consider to be one of the main contributions of this study. Tabs. 6 and 7 summarize the results obtained after five runs of GA-SFLA based on the internal measures of the GA for feature selection and SFLA for text document clustering, respectively. Similar scenarios to those in Tabs. 5 and 6 can also be observed in Tabs. 6 and 7, respectively.

Table 6: Results of multiple runs on the entire dataset with different numbers of features selected based on the GA fitness value

Dataset size	Number of features selected	Average fitness of five runs	Best fitness of five runs	Average processing time of five runs in s
	20%	22678.84879	23032.2269	15889.05864
Entire dataset (18846)	50%	49355.99738	53275.2761	28804.1806
Average		**36017.42309**	**38153.7515**	22346.61962

Table 7: Results of multiple runs of the GA-SFLA on the entire dataset based on the inter-cluster and intra-cluster similarity ratios

Dataset size	Number of features selected	Average fitness of five runs	Best fitness of five runs	Average processing time of five runs in s
Entire dataset (18846)	20%	42495.0849	198066.3855	81532.2441
	50%	146069.6592	713778.7629	103248.1564
Average		**94282.3721**	**455922.5742**	92390.200**3**

6.3 Experiment (3)

In this experiment, we used the same datasets to compare GA-SFLA with the classical K-means clustering algorithm. The experiments were run five times for each algorithm. Tab. 8 shows the results for both algorithms on 20Newsgroup datasets of size 500, 1000, 1500, and 2000, with 50% of the number of features selected in each dataset. Moreover, based on the corpus used, the number of clusters for both algorithms was set in advance to 20. In the comparison, we used the same feature selection technique in both approaches (i.e., the GA) and implemented classical K-means using a predefined function in Python.

Tab. 8 presents for each algorithm, the average F-micro and F-macro values for five runs after the GA feature selection method was applied, where the best results are highlighted in bold. Furthermore, the last column in the table (Gap%) indicates the percent difference between the results obtained by GA-K-means (worst results) and GA-SFLA (best results).

It can be seen that both the F-micro and F-macro values for the proposed algorithm are better than those for the GA-K-means algorithm. Furthermore, the last column shows that GA-SFLA improved the average of F-micro by 62% and the average of F-macro by 70%

(a negative value indicates that the results of GA-K-means is lower than that of GA-SFLA). In contrast, the processing time of the proposed algorithm is worse than that of GA-K-means. Therefore, we can conclude that proposed algorithm outperforms the other method, particularly in the case of large numbers of documents. This is because the SFLA is better in terms of search precision owing to the local and global messaging exchange. Moreover, the SFLA can learn from past experience.

Table 8: Comparison between GA-SFLA and GA-K-means

	GA-SFLA		**GA-K-means**		**Gap%**	
Dataset size	F-micro	F-macro	F-micro	F-macro	F-micro	F-macro
500	**0.2104**	**0.2013**	0.0816	0.0601	**-61%**	**-70%**
1000	**0.1956**	**0.1839**	0.0708	0.0551	**-63%**	**-70%**
1500	**0.2356**	**0.2251**	0.0809	0.0562	**-65%**	**-75%**
2000	**0.2275**	**0.2102**	0.0848	0.0655	**-62%**	**-68%**
Average	**0.2172**	**0.2051**	0.0795	0.0592	**-62%**	**-70%**

6.4 Experiment (4)

As in the case of Experiment (3), the purpose of Experiment (4) is to assess and compare GA-SFLA and the GA-K-means clustering solutions, but for the entire text document dataset. The average scores of the external measures and processing time (in s) in five runs were compared to find the most effective algorithm, as shown in Tabs. 9 and 10. As before, the last column (Gap%) in Tab. 10 indicates the percent difference between the results obtained by GA-K-means (worst results) and GA-SFLA (best results). A similar scenario to that in Tab. 8 can also be observed in Tabs. 9 and 10.

Table 9: GA-SFLA applied to the entire dataset

	GA-SFLA		
Dataset size	F-micro	F-macro	Processing time (average of five runs in s)
Entire dataset (18846)	**0.18076**	**0.16528**	103290.9981

Table 10: GA-K-means applied to the entire dataset

	GA-K-means			**Gap%**	
Dataset size	F-micro	F-macro	Processing time (average of five runs in s)	F-micro	F-macro
Entire dataset (18846)	0.1308	0.1309	**34755.6566**	**-27%**	**-20%**

It can be seen that the F-micro and F-macro values indicate that GA-SFLA is slightly superior to GA-K-means, whereas the processing time is higher for GA-SFLA. The reason for obtaining only slightly better results may be that the entire dataset is well clustered in definite categories, which simplifies the task of the K-means algorithm. However, when the data sets were randomly selected, as in Experiment (3), GA-SFLA could identify the clusters better than the K-means algorithm. That is, GA-SFLA appears to perform better than K-means when the data are not well clustered.

6.5 Experiment (5)

The main goal of this experiment is to assess the benefit of feature selection in text document clustering. We recall that feature selection is responsible for extracting topic-related terms that may facilitate the presentation of the content of each document. Tabs. 11, 12, 13 and 14 compare the results obtained by running GA-SFLA and SFLA only (without feature selection), which implies that all features in the lexicon were used for clustering**.** Tabs. 11, 12, 13 and 14 present the average and the best fitness values in five runs on 20Newgroup datasets with different sizes: 500, 1000, 1500, 2000 (Tabs. 11 and 12), and the entire text document dataset (Tabs. 13 and 14). In addition, the average processing time is calculated in second. The last column (Gap%) in Tabs. 12 and 14 indicates the percent difference between the best results obtained by the version of the algorithm that does not include feature selection (worst results) and the results obtained using feature selection (best results).

Table 11: Results of SFLA clustering with feature selection

	GA-**SFLA**		
Dataset size	Average fitness of five runs	Best fitness of five runs	Average processing time of five runs in s
500	117.5687	298.5155	1893.5787
1000	341.3758	1030.4602	10059.2930
1500	519.7878	1605.8567	10871.3446
2000	599.7699	1271.0224	18845.2108
Average	**394.6255**	**1051.4637**	10417.3567

Table 12: Results of SFLA clustering without feature selection

	SFLA only (without feature selection)			**Gap%**
Dataset size	Average fitness of five runs	Best fitness of five runs	Average processing time of five runs in s	
500	57.8811	176.4513	679.3026	-40%
1000	123.8652	343.6142	1333.21073	-66%
1500	217.5755	567.1643	1964.1846	-64%
2000	483.7157	905.8840	2620.9877	-28%
Average	220.7593	498.2784	**1649.4214**	**-49%**

Table 13: Results of SFLA clustering with feature selection applied to the entire dataset

	GA-SFLA		
Dataset size	Average fitness of five runs	Best fitness of five runs	Average processing time of runs in s
Entire Dataset	**146069.6592**	**713778.7629**	103248.1564

Table 14: Results of SFLA clustering without feature selection applied to the entire dataset

	SFLA only (without feature selection)			**Gap%**
Dataset size	Average fitness of five runs	Best fitness of five runs	Average processing time of five runs in s	
Entire **Dataset**	105396.1954	268192.5573	**27173.38541**	**-62%**

It can be clearly seen that the feature selection phase improved the text document clustering results in all test cases. This indicates that the GA is an effective method for finding the most informative feature set among a large number of features, thus improving text document clustering. This is particularly helpful when only a small number of dimensions are relevant to certain clusters, which is the case in almost all text document clustering problems. Fig. 7 shows the average intra-cluster and inter-cluster similarity results of five runs between GA-SFLA and SFLA. Furthermore, Fig. 8 shows the processing time for both algorithms.

Figure 7: GA-SFLA and SFLA intra-cluster *vs.* inter-cluster similarity results

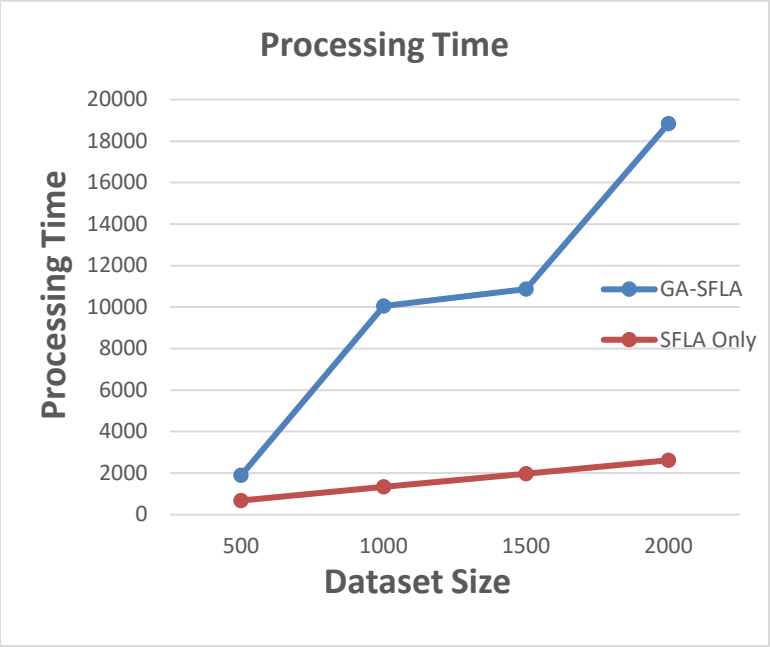

Figure 8: GA-SFLA and SFLA processing time

In conclusion, owing to the irrelevant dimensions that generate a large amount of noise and mask the actual clusters, feature selection is an important step to enhance the clustering process. However, the average processing time of the proposed algorithm,

compared with the average of processing time of the SFLA without feature selection, is high. This is obviously due to the additional processing time required by the GA to select the best features.

7 Conclusions and future work

The World Wide Web contains a wealth of textual data that can be mined to extract useful information for various real-life applications. However, the large amounts of these data render this process a considerable challenge. Consequently, there is a continuous need for effective techniques that are capable of automatically identifying useful information from textual data. Data clustering is an important technique in this category; it is an unsupervised learning mechanism that aims to cluster objects into subsets, so that each subset shares common characteristics.

In this study, we proposed a new text document clustering method that utilizes two meta-heuristic algorithms for enhanced clustering. The proposed GA-SFLA method combines a GA and an SFLA; the GA is utilized for feature selection, whereas the SFLA performs clustering. We conducted multiple experiments involving the 20Newsgroup dataset, obtained from the UCI Machine Learning Repository, to test the proposed approach. Overall, the results were thoroughly assessed by applying the most commonly used measures for text clustering, namely, the internal (maximizing the ratio of intra-cluster to inter-cluster similarity) and the external (F-micro and F-macro) evaluation measures.

The results of GA-SFLA were compared with those of the classical K-means clustering algorithm in terms of the external measures for different document sizes. For all tested datasets, the results indicated the efficiency of the proposed approach in achieving significantly better clustering results than the K-means. Moreover, we compared the proposed approach with another evolutionary clustering algorithm (SFLA) without the feature selection phase in terms of the clustering quality measure. Again, the experimental results demonstrated the efficiency of the proposed approach for all tested datasets; this indicates the importance of feature selection. Overall, considering the results of the experiments, it can be concluded that using GA-SFLA on the 20Newsgroup dataset can greatly enhance the text document clustering process. Nevertheless, this requires a long computational time.

For future work, it is intended that this experiment be applied to other widely used text datasets, such as Reuters-21578. Furthermore, another feature selection evaluation technique, such as the wrapper method, may be used. In addition, we will attempt to improve the cohesiveness of the resulting clusters and optimize processing time. Moreover, other representation methods in the GA, such as integer representation, as well as different crossover and mutation operators can be considered.

Acknowledgement: This research was supported by a grant from the Research Center of the Center for Female Scientific and Medical Colleges Deanship of Scientific Research, King Saud University. Also, the authors thank the Deanship of Scientific Research and RSSU at King Saud University for their technical support.

References

Abualigah, L. M.; Khader, A. T.; Al-Betar, M. A. (2016): Unsupervised feature selection technique based on genetic algorithm for improving the text clustering. *7th International Conference on Computer Science and Information Technology*, pp. 1-6.

Aggarwal, C.; Reddy, K. (2014): *Data Clustering: Algorithms and Applications*. Chapman and Hall/CRC, Boca Raton.

Al-Jadir, I.; Wong, K. W.; Fung, C. C.; Xie, H. (2017): Text dimensionality reduction for document clustering using hybrid memetic feature selection. *International Workshop on Multi-Disciplinary Trends in Artificial Intelligence*, pp. 281-289.

Al-Malak, S.; Hosny, M. (2016): A multimodal adaptive genetic clustering algorithm. *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, pp. 1453-1454.

Alsaeedi, A.; Fattah, M. A.; Aloufi, K. (2017): A hybrid feature selection model for text clustering. *7th IEEE International Conference on System Engineering and Technology*, pp. 7-11.

Amiri, B.; Fathian, M.; Maroosi, A. (2009): Application of shuffled frog-leaping algorithm on clustering. *International Journal of Advanced Manufacturing Technology*, vol. 45, no. 1-2, pp. 199-209.

Asuncoin, A.; Newman, D. (2007): *UC Irvine Machine Learning Repository*. https://archive.ics.uci.edu/ml/index.php.

Bhaduri, A.; Bhaduri, A. (2009): Color image segmentation using clonal selection-based shuffled frog leaping algorithm. *International Conference on Advances in Recent Technologies in Communication and Computing*, pp. 517-520.

Binitha, S.; Sathya, S. S. (2012): A survey of bio inspired optimization algorithms. *International Journal of Soft Computing and Engineering*, vol. 2, no. 2, pp. 137-151.

Cummins, R.; O'Riordan, C. (2006): Evolved term-weighting schemes in information retrieval: an analysis of the solution space. *Artificial Intelligence Review*, vol. 26, no. 1-2, pp. 35-47.

Davies, D. L.; Bouldin, D. W. (1979): A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 2, pp. 224-227.

Dhillon, I.; Kogan, J.; Nicholas, C. (2004): *Feature Selection and Document Clustering. Survey of Text Mining.* Springer, New York.

Elbeltagi, E.; Hegazy, T.; Grierson, D. (2005): Comparison among five evolutionary-based optimization algorithms. *Advanced Engineering Informatics*, vol. 19, no. 1, pp. 43-53.

Eusuff, M.; Lansey, K.; Pasha, F. (2006): Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. *Engineering Optimization*, vol. 38, no. 2, pp. 129-154.

Eusuff, M. M.; Lansey, K. E. (2003): Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources Planning and Management*, vol. 129, no. 3, pp. 210-225.

Fang, Y.; Yu, J. (2011): Application of shuffled frog-leaping algorithm in web's text cluster technology. *International Conference on Web Information Systems and Mining*, pp. 363-368.

Goldberg, D. E. (1989): *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Halkidi, M.; Batistakis, Y.; Vazirgiannis, M. (2001): On clustering validation techniques. *Journal of Intelligent Information Systems*, vol. 17, no. 2-3, pp. 107-145.

Han, J.; Kamber, M.; Pei, J. (2006): *Data Mining: Concepts and Techniques.* Morgan Kaufmann, Amsterdam; Boston: San Francisco, CA.

Holland, J. H. (1975): *Adaptation in Natural and Artificial Systems.* MIT Press Cambridge, MA, USA.

Hong, S.; Lee, W.; Han, M. (2015): The feature selection method based on genetic algorithm for efficient of text clustering and text classification. *International Journal of Advances in Soft Computing and Its Applications*, vol. 7, no. 1, pp. 2074-8523.

Hosny, M.; Hinti, L.; Al-Malak, S. (2018): Co-evolutionary framework for adaptive multidimensional data clustering. *Intelligent Data Analysis*, vol. 22.

Hruschka, E. R.; Campello, R. J.; Freitas, A. A.; Carvalho, A. (2009): A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems*, vol. 39, no. 2, pp. 133-155.

Huang, A. (2008): Similarity measures for text document clustering. *Proceedings of the Sixth New Zealand Computer Science Research Student Conference*, pp. 9-56.

Iglesias, F.; Kastner, W. (2013): Analysis of similarity measures in times series clustering for the discovery of building energy patterns. *Energies*, vol. 6, no. 12, pp. 579-597.

Jagatheeshkumar, G.; Brunda, S. S. (2017): An analysis of efficient clustering methods for estimates similarity measures. *4th International Conference on Advanced Computing and Communication Systems*, pp. 1-3.

Janani, R. (2016): Genetic algorithm based document clustering. *International Journal of Advanced Research Trends in Engineering and Technology*, vol. 3, no. 20.

Jong, K. (1975): *Analysis of the Behavior of a Class of Genetic Adaptive Systems.* University of Michigan Ann Arbor, MI, USA.

Kalashami, S.; Chabok, S. J. (2016): Use of the improved frog-leaping algorithm in data clustering. *Journal of Computer & Robotics*, vol. 9, no. 2, pp. 19-26.

Karakoyun, M.; Babalik, A. (2015): Data clustering with shuffled leaping frog algorithm (SFLA) for classification. *International Conference on Intelligent Computing, Electronics Systems and Information Technolog*, pp. 25-26.

Karol, S.; Mangat, V. (2013): Evaluation of text document clustering approach based on particle swarm optimization. *Open Computer Science*, vol. 3, no. 2.

Kaufman, L.; Rousseeuw, P. (1987): Clustering by means of medoids. https://www.researchgate.net/publication/243777819_Clustering_by_Means_of_Medoids.

Kaur, P.; Rohil, H. (2015): Swarm intelligence in data clustering: a comprehensive review. *International Journal of Research in Advent Technology*, vol. 3, no. 4.

Lang, K. (1995): Newsweeder: learning to filter netnews. *Machine Learning Proceedings*, pp. 331-339.

Lewis, D. D. (1987): *Reuters-21578 Text Categorization Collection Data Set*.

http://archive.ics.uci.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection.

Liu, B. (2011): *Web Data Mining.* Morgan Kaufmann, Amsterdam; Boston: San Francisco, CA.

Liu, L.; Kang, J.; Yu, J.; Wang, Z. (2005): A comparative study on unsupervised feature selection methods for text clustering. *International Conference on Natural Language Processing and Knowledge Engineering*, pp. 597-601.

Mitchell, T. (1999): *Twenty Newsgroups Data Set.*

http://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups.

Porter, M. (1980): An algorithm for suffix stripping. *Program*, vol. 14, pp. 130-137.

Patil, L. H.; Atique, M. (2013): A novel approach for feature selection method TF-IDF in document clustering. *3rd IEEE International Advance Computing Conference*, pp. 858-862.

Ramya, G.; Chandrasekaran, M. (2013): An efficient heuristics for minimizing total holding cost with no tardy jobs in job shop scheduling. *International Journal of Computer and Communication Engineering*, vol. 2, no. 2, pp. 216.

Rao, R. V.; Savsani, V. J. (2012): *Mechanical Design Optimization Using Advanced Optimization Techniques.* Springer Verlag, London.

Salton, G.; Buckley, C. (1988): Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, vol. 24, no. 5, pp. 513-523.

Santra, A. K.; Christy, C. J. (2012): An efficient document clustering by optimization technique for cluster optimality. *International Journal of Computer Applications*, vol. 43, no. 16, pp. 15-20.

Shen, T.; Nagai, Y.; Gao, C. (2019): Optimal building frame column design based on the genetic algorithm. *Computers, Materials & Continua,* vol. 58, no. 3, pp. 641-651.

Sun, X.; Wang, Z.; Zhang, D. (2008): A web Document classification msethod based on shuffled frog leaping algorithm. *Second International Conference on Genetic and Evolutionary Computing*, pp. 205-208.

Torres, G. J.; Basnet, R. B.; Sung, A. H.; Mukkamala, S.; Ribeiro, B. M. (2009): A similarity measure for clustering and its applications. *International Journal of Electrical and Computer Engineering*, vol. 3, no. 3, pp. 164-170.

Wang, C.; Song, Y.; Li, H.; Zhang, M.; Han, J. (2018): Unsupervised meta-path selection for text similarity measure based on heterogeneous information networks. *Data Mining and Knowledge Discovery*, vol. 32, no. 6, pp. 1735-1767.

Wang, M.; Di, W. (2010): A modified shuffled frog leaping algorithm for the traveling salesman problem. *Sixth International Conference on Natural Computation*, vol. 7. pp. 3701-3705.

Willett, P. (2006): The porter stemming algorithm: then and now. *Program*, vol. 40, no. 3, pp. 219-223.

Xu, R.; WunschII, D. (2005): Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645-678.

Zhang, R.; Liu, C.; Liang, S.; Zhang, X.; Dong, W. et al. (2016): An improved clustering algorithm based on multi-swarm intelligence. *International Symposium on Computer, Consumer and Control*, pp. 489-492.