

Article

# Swarm Intelligence Algorithms in Text Document Clustering with Various Benchmarks

Suganya Selvaraj <sup>1</sup>  and Eunmi Choi <sup>1,2,\*</sup>

<sup>1</sup> Department of Financial Information Security, Kookmin University, Seoul 02707, Korea; suganya@kookmin.ac.kr

<sup>2</sup> Department of Software, College of Computer Science, Kookmin University, Seoul 02707, Korea

\* Correspondence: emchoi@kookmin.ac.kr; Tel.: +82-2-910-4790; Fax: +82-2-910-4868

**Abstract:** Text document clustering refers to the unsupervised classification of textual documents into clusters based on content similarity and can be applied in applications such as search optimization and extracting hidden information from data generated by IoT sensors. Swarm intelligence (SI) algorithms use stochastic and heuristic principles that include simple and unintelligent individuals that follow some simple rules to accomplish very complex tasks. By mapping features of problems to parameters of SI algorithms, SI algorithms can achieve solutions in a flexible, robust, decentralized, and self-organized manner. Compared to traditional clustering algorithms, these solving mechanisms make swarm algorithms suitable for resolving complex document clustering problems. However, each SI algorithm shows a different performance based on its own strengths and weaknesses. In this paper, to find the best performing SI algorithm in text document clustering, we performed a comparative study for the PSO, bat, grey wolf optimization (GWO), and K-means algorithms using six data sets of various sizes, which were created from BBC Sport news and 20 newsgroups. Based on our experimental results, we discuss the features of a document clustering problem with the nature of SI algorithms and conclude that the PSO and GWO SI algorithms are better than K-means, and among those algorithms, the PSO performs best in terms of finding the optimal solution.

**Keywords:** swarm intelligence algorithms; text document clustering; artificial intelligence; data mining



**Citation:** Selvaraj, S.; Choi, E. Swarm Intelligence Algorithms in Text Document Clustering with Various Benchmarks. *Sensors* **2021**, *21*, 3196. <https://doi.org/10.3390/s21093196>

Academic Editors: Gianvito Pio, Roberto Corizzo and Michelangelo Ceci

Received: 31 March 2021

Accepted: 29 April 2021

Published: 4 May 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 is the application of cluster analysis referring to the unsupervised classification of textual documents into clusters based on content similarity. Text document clustering can be applied in organizing large document collections, extracting hidden information from data generated by IoT sensors, finding similar documents, detecting duplicate content, search optimization, and recommendation systems [1,2]. These text document sources may come from web pages, blog posts, news articles, or other text files [3]. Extracting relevant information from the data is a challenging task that needs fast and high-quality document clustering algorithms. The most popular algorithms for clustering are K-means and its variants such as bisecting K-means and K-medoids [4]. The K-means algorithm is a simple, fast, and unsupervised partitioning algorithm offering easily parallelized and intuitive results [5].

Swarm intelligence (SI) is the collective behavior of self-organized and decentralized systems, which includes simple and unintelligent individuals that follow simple rules or behaviors to accomplish very complex tasks with limited local information. SI algorithms have been discovered from observing natural or artificial behaviors such as bird flocks, fish schools, and ant food foraging [6]. SI algorithms include particle swarm optimization (PSO), the bat algorithm (BA), grey wolf optimization (GWO), the firefly algorithm (FFA), ant colony optimization (ACO), the artificial fish swarm algorithm (AFSA), and artificial bee colony optimization (ABC) [7]. These algorithms can be used in many real-world problems such as traveling salesman problems (TSPs), feature selection (FS), robot swarm

learning, cluster analysis, and scheduling. SI algorithms have advantages like adaptability, robustness, scalability, and flexibility over traditional clustering algorithms [8]. These characteristics of SI algorithms make them suitable for solving complex text document clustering problems.

However, each SI algorithm shows a different performance based on its own strengths and weaknesses, which necessitates a comparative evaluation of the SI algorithms to find the best one that offers the optimal solution. In this paper, therefore, we perform a comparative study of PSO, BA, GWO, and K-means using six data sets of various sizes which were created from BBC Sport [9] news and 20 newsgroups [10] to find the best performing SI algorithm in text document clustering. Here, the text documents were preprocessed using natural language processing, and different metrics such as purity, homogeneity, completeness, V-measure, ARI, and average running time were used for the comparison [7]. Based on experimental results, we discuss the features of a document clustering problem with the nature of SI algorithms and conclude that PSO and GWO are better than the traditional K-means clustering algorithm and PSO is the best performing algorithm in terms of finding the optimal solution.

## 2. Related Work

This section summarizes the previous comparative study results for SI algorithms in a few applications such as TSP, FS, robot swarm learning, scheduling, clustering, and document clustering.

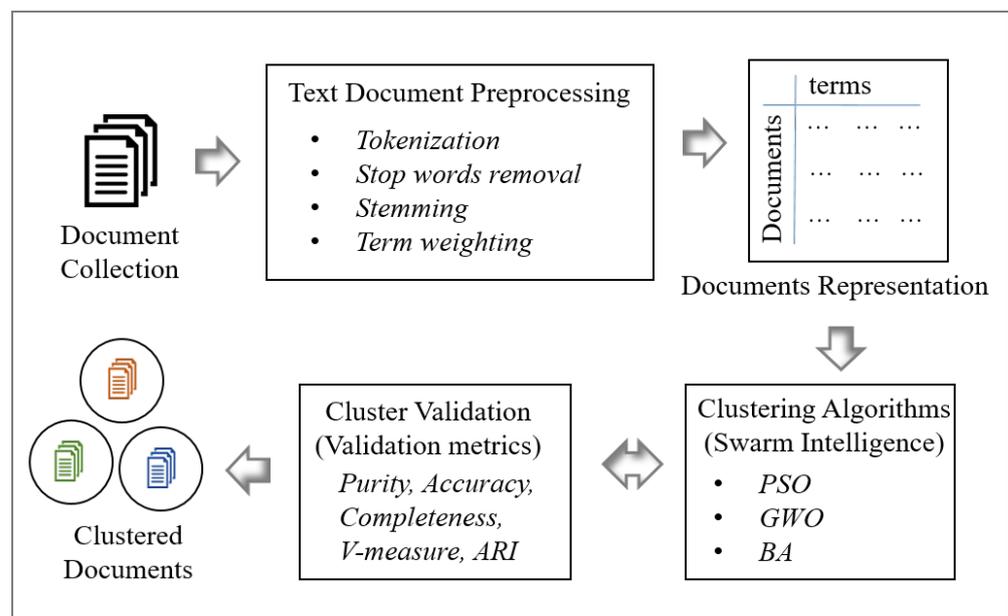
Shima Sabet and Farokhi [11] implemented the TSP with SI algorithms such as the genetic algorithm (GA), PSO, ABC, and ACO, and the performance of these algorithms was compared. This study states that ABC shows better performance than other algorithms and also ACO shows good performance for less than 80 cities in TSP. A study by Basir and Ahmad [12] compared a few SI algorithms for FS and suggests that SI needs to be studied more for FS. Fan et al. [13] performed a comparative study for PSO, GWO, and BA in robot swarm learning and their result shows that PSO outperforms BA and GWO. In addition, GWO performs better than BA and PSO for a large number of robots and a long communication range. S.J.Mohana [14] reported a comparative analysis study for cloud scheduling for PSO and ACO algorithms and the study shows that PSO performs better than ACO. Elhady and Tawfeek [15] compared ABC, PSO, and ACO in dynamic cloud task scheduling and the results show that ABC outperforms PSO and ACO. Ref. Zhu et al. [16] compared twelve SI algorithms for an uninhabited combat air vehicle path-planning problem and their results show that the spider monkey optimization (SMO) algorithm performs better than others in discovering a safe path.

The SI algorithms FFA, CS, BA, and PSO were compared in medical data clustering and results revealed that CS performs slower than others and PSO and BA are relatively faster than others [17]. Figueiredo et al. [18] studied the comparison of SI algorithms in the clustering domain and concluded that PSO is the best performing algorithm. Algorithms ABC, ACO, and AFSA are the second-level performers. Additionally, SI algorithms such as the whale optimization algorithm (WOA) and BA, and metaheuristic algorithm differential evolution (DE), were combined with the proposed method OpStream for data stream clustering and these results were compared with the state-of-art algorithms DenStream and CluStream [19–21]. Lu et al. [22] proposed the text clustering PSO (TCPSO) algorithm by extending the PSO algorithm to solve the variable weighing in text clustering and compared these results with a few other algorithms such as bisection K-means and K-means. Judith and Jayakumari [23] proposed a hybrid algorithm that includes K-means and PSO to solve the distributed document clustering problem. Abualigah et al. [24] compared a few nature-inspired optimization algorithms in text document clustering and the result showed that, according to the accuracy measure and F-measure, GWO is the best performing algorithm and GA is the least performing algorithm. Rashaideh et al. [25] studied the GWO algorithm in a text document clustering problem and their result shows that GWO performs better than the  $\beta$ -hill climbing and hill-climbing algorithms.

### 3. Process of Text Document Clustering

This section describes the main processes of text document clustering such as text document preprocessing, document representation, clustering, and cluster validation.

As shown in Figure 1, preprocessing converts all given documents into a document matrix. Then, algorithms such as K-means and SI algorithms (PSO, GWO, and BA) are used to create text document clusters by using the document matrix. A cluster validation step is used to validate the clusters and performance of the clustering techniques.



**Figure 1.** Process of text document clustering.

#### 3.1. Text Document Preprocessing

The preprocessing step greatly influences the result of clustering. Here, we applied a preprocessing step to convert the set of documents into a mathematical data model with which the computer can deal [26,27]. The result of this preprocessing is a  $m \times n$  term-document matrix. This matrix is built with the table of frequencies and occurrences of terms in each document. Here,  $m$  is the number of unique terms in the document collection and  $n$  is the number of documents in the collection. We use the techniques of tokenization, stop word removal, stemming, and term weighting involved in the preprocessing as follows [26].

- **Tokenization:** Tokenization is a process that splits a stream of text documents into words or terms and removes empty sequences. Here, each word or symbol is taken from the first character to the last character, where each word is called a token [28]. However, sometimes defining a “word” is difficult. A tokenizer often depends on simple heuristics such as [29]:
  - the resulting list of tokens may or may not include punctuation and white space;
  - tokens are separated by white space characters such as spaces, line breaks, or punctuation characters.
- **Stop word removal:** Stop words such as an, are, for, be, and other common words are more frequent and short functional words also take small weighting. These words should be removed from the document to increase the performance of text document clustering. The list of stop words is available at <http://members.unine.ch/jacques.savoy/clef/index.html> (accessed on 22 April 2021) and includes 571 words [28].
- **Stemming:** Stemming is the process of reducing inflectional words to the same root by removing the affixes (prefixes and suffixes) of each word. For example [26,29]:
  - section, dissect, and intersect all have a common origin or source Sect called the feature;

- if the word ends with *ed* or *ing* or *ly*, we remove the *ed* or *ing* or *ly*, respectively.
- Term weighting: The term weighting is assigned for each term or feature by considering the frequency of each term in the document. The term frequency-inverse document frequency (TF-IDF) is widely used in weighting methods. Each document is represented as a vector of term weights as shown in Equation (1) [28].

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

The term weight for the feature  $j$  in document  $i$  is calculated using the following Equation (2).

$$w_{i,j} = tf(i,j) \times idf(i,j) = tf(i,j) \times \log\left(\frac{n}{df(j)}\right) \quad (2)$$

In Equation (2),  $w_{i,j}$  represents the weight of document  $i$  and term  $j$ .  $tf(i,j)$  denotes the occurrences of term  $j$  in document  $i$ .  $idf(i,j)$  is the inverse document frequency.  $n$  represents the number of all documents in the data set and  $df(j)$  represents the number of documents that contain feature  $j$ . Usually, the vector space model (VSM) is used in text mining to represent the document features as a vector (row) of weight. Equation (3) is the common format of the VSM and that includes  $n$  documents and each document includes  $t$  terms.

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

### 3.2. General Process of SI Algorithms in Text Document Clustering

This section describes the general process of SI algorithms in text document clustering [30].

As shown in Figure 1, the text document clustering algorithms directly use the document matrix to convert the set of document data sets into meaningful sub-collections [28]. Some common fundamental phases of SI algorithms are 1. initialize the population of swarm agents; 2. define the termination condition; 3. create document clusters using agents' positions (centroids); 4. evaluate the fitness function; 5. agent communication; 6. update and move agents; and 7. return the global best solution.

Before the initialization of the algorithm, the values of the algorithm parameters should be defined. Each algorithm has different parameters and those parameters may affect the result of the algorithm. The evolutionary process of the SI algorithm starts with initialization. In the second phase, the termination condition needs to be defined to stop the execution of the algorithm. Here, we use the number of iterations as the stopping criterion. In this experiment, we used different iteration values (1, 10, 20, ..., 90, and 100) to evaluate and compare the performance of algorithms with different iterations. In the third phase, we create clusters by assigning each document to the closest center (agent's position) by calculating the distance to each centroid. The fourth step evaluates the fitness function that is responsible for the evaluation of search agents. The fitness function is either single or combined. Usually, one of the metrics is used as a fitness function. In this study, we use the purity metric as a fitness function. The highest value of purity is considered the best fitness solution. Here, agent communication is used to share information among agents. For example, in the PSO algorithm, agents share the personal best solutions among agents. Agents in an SI algorithm update and move based on some mathematical background of the algorithm. In the last step, after meeting the stopping criterion the SI algorithm returns the best fitness solution from the search agent.

### 3.3. Clustering Evaluation Metrics

Cluster evaluation is a post-clustering technique that validates the quality of final results from clustering algorithms. There are two kinds of validating criteria available: external and internal. External criteria measure the performance by matching clustering structure to a priori knowledge, which includes metrics such as purity, entropy, F-measure, homogeneity, accuracy, completeness, V-measure, and adjusted rand index (ARI). In this study, we used external evaluation metrics such as purity, homogeneity, completeness, V-measure, and ARI to measure the quality of text document clustering algorithms as shown in Figure 1.

#### 3.3.1. Purity

Purity measures whether the clusters contain documents from a single category. The purity value ranges between 0 to 1 and the purity value of an ideal cluster is 1. The purity value is computed by dividing the number of correctly assigned documents by  $N$  and the formula is defined as follows [31,32]:

$$Purity(\Omega, C) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j| \quad (4)$$

where  $\Omega = \{\omega_1, \omega_2, \dots, \omega_j\}$  is the set of clusters and  $C = \{c_1, c_2, \dots, c_j\}$  is the set of classes.

#### 3.3.2. Homogeneity, Completeness, and V-Measure

Both homogeneity and completeness range between 0 and 1. Homogeneity is 1 if all of its clusters contain only data points that are members of a single class. Completeness is 1 if all the data points that are members of a given class are elements of the same cluster.

Homogeneity (Equation (5)) and completeness (Equation (6)) values can be calculated as follows [33,34]:

$$h = 1 - \frac{H(C|K)}{H(C)} \quad (5)$$

$$c = 1 - \frac{H(K|C)}{H(K)} \quad (6)$$

where  $H(C|K)$  is the conditional entropy of the classes defined as:

$$H(C|K) = - \sum_{c=1}^{|C|} \sum_{k=1}^{|K|} \frac{n_{c,k}}{n} \times \log \left( \frac{n_{c,k}}{n_k} \right) \quad (7)$$

and  $H(C)$  is the entropy of the classes defined as:

$$H(C) = - \sum_{c=1}^{|C|} \frac{n_c}{n} \times \log \left( \frac{n_c}{n} \right) \quad (8)$$

Here,  $n$  is the total number of samples.  $n_c$  and  $n_k$  are the number of samples belonging to class  $c$  and cluster  $k$ , respectively.  $n_{c,k}$  is the number of samples from class  $c$  assigned to cluster  $k$ .

V-measure can be defined as the harmonic mean of homogeneity and completeness as follows [33,34]:

$$v = 2 \times \frac{h \times c}{h + c} \quad (9)$$

#### 3.3.3. ARI

The rand index (RI) measures the similarity between two data clusters by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusters.

The ARI value is adjusted for the chance of raw RI and can be defined as follows [34,35]:

$$ARI = (RI - ExpectedRI) / (max(RI) - ExpectedRI) \quad (10)$$

The ARI ranges between 0 and 1. The value of ARI is 1 if the clusters are identical.

#### 4. Algorithms for Text Document Clustering

This section shows the modified structure of K-means and SI algorithms such as PSO, BA, and GWO to perform document clustering.

##### 4.1. K-Means Clustering Algorithm

K-means is a simple and iterative algorithm that clusters a group of data sets into a K predefined number of clusters [36]. Algorithm 1 shows a basic K-means clustering algorithm.

The K-means algorithm starts with random initialization of cluster centers. In step 2, the algorithm assigns each document into its closest centroid based on their similarity. Step 3 recalculates the centroid of each cluster. Steps 2 and 3 will repeat they until meet the termination condition.

---

##### Algorithm 1: K-means Document Clustering Algorithm

---

- Step 1. Randomly select K points as the initial cluster centers.
  - Step 2. Assign all points (documents) to the closest centroid.
  - Step 3. Recalculate the centroid of each cluster.
  - Step 4. Repeat steps 2 and 3 until termination condition is reached.
- 

##### 4.2. SI Algorithms for Document Clustering

###### 4.2.1. PSO Algorithm

The PSO algorithm is an optimization algorithm developed based on bird and fish flock movement behavior. The PSO algorithm aims to locate all the particles in the optimal position in a multi-dimensional space [36]. In this study, each particle in the PSO algorithm represents a cluster centroid of text document clusters. The initialization phase is used to initialize the number of clusters K and a few PSO parameters. After the initialization, new positions (Equation (11)) and velocities (Equation (12)) are calculated for each particle. Each position is considered as a centroid and clusters are created by assigning each document to the closest center (particle's position) by calculating the distance to each centroid. Then, the fitness function evaluates the clusters, and the global best (gbest) and individual best (pbest) solutions are updated until they reach the termination condition criteria as shown in Algorithm 2. As a result, the algorithm finally returns the global best solution [37].

$$V_{id}^{k+1} = \omega * v_{id}^k + c_1 r_1^k (pbest_{id}^k - x_{id}^k) + c_2 r_2^k (gbest_d^k - x_{id}^k) \quad (11)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad (12)$$

In Equations (11) and (12),  $V_{id}^k$  and  $x_{id}^k$  stand for the speed and position of the particle  $i$  at iteration  $k$ .  $pbest_{id}^k$  is the personal best position of  $i$ .  $gbest_d^k$  is the global best position among all individuals in the  $d$ -dimension.  $\omega$  is the inertia weight that balances the global search and local search abilities.  $c_1$  and  $c_2$  are positive constants used to control the speed of the particle.  $r_1$  and  $r_2$  are random parameters within  $[0, 1]$  [31,36].

**Algorithm 2:** PSO document clustering algorithm**Result:** Global best (gbest) position**Initialization;**

- Randomly choose K cluster centers to initialize a population of particles (centroids).
- Initialize position and velocity.

**while** *Until the stop criteria are met* **do**    **while** *For each particle* **do**

- Assign each document to the closest center (particle's position) by calculating the distance to each centroid (clustering).
- Compute the fitness value.
- Update personal best (pbest) and global best (gbest) positions.
- Update the velocity and position of the particle to generate new solution (Equation (11) and Equation (12)).

**end****end****return** *gbest position*

## 4.2.2. BA Algorithm

The BA algorithm is developed based on the echolocation behavior of microbats. This algorithm follows some idealized rules as follows. 1. All bats use echolocation to sense the distance of prey or predator and also find the barriers in the path. 2. Bats fly with a velocity  $v_i$  and position  $x_i$ ; they have loudness  $A_i$  and frequency  $F$  to reach their prey and adjust the frequency of pulse emission  $r$ . 3. As they get close to the prey, loudness decreases ( $A$ ) and pulse increases ( $r$ ) [6].

$$F_i + (F_{max} - F_{min})\beta \quad (13)$$

$$V_i^t = V_i^{t-1} + (X_i^t - X^*)F_i \quad (14)$$

$$X_i^t = X_i^{t-1} + V_i^t \quad (15)$$

$$X_{new} = X_{old} + \epsilon A^t \quad (16)$$

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)], A_i^{t+1} \quad (17)$$

In this study, a bat's position in the BA algorithm represents a cluster centroid of text document clusters. The initialization phase used to initialize the number of clusters K and a few BA parameters is shown in Algorithm 3. After the initialization, Equations (13)–(17) are used to search for the new positions. The centroids and clusters are created by assigning each document to the closest center (particle's position) by calculating the distance to each centroid. Then, the fitness function evaluates the clusters and the global best (gbest) solution is updated until it reaches the stopping criteria. As a result, the algorithm finally returns the global best solution.

Here,  $Q$  is the frequency band parameter,  $\beta$  is a random vector in a uniform distribution between 0 and 1.  $\epsilon$  is a random walk in the normal distribution between  $-1$  and  $1$ . Bats adjust their position according to Equations (13)–(15). Equation (16) is used to find the local search using a random walk. Equation (17) is used to find the local search control parameter  $r_i$  and loudness  $A$  [38,39].

**Algorithm 3:** BA document clustering algorithm**Result:** Global best position**Initialization;**

- Randomly choose K cluster centers to initialize a population of n bats' positions.
- Initialize position, velocity, and other parameters for each bat.

**while** *Until the stop criteria are met* **do**  **while** *For each bat* **do**

Generate new solutions by adjusting frequency (Equation (13)).

- Update the velocity (Equation (14)).
- Update the position (Equation (15)).

**if** *rand* >  $r_i$  **then**

Update the position (Equation (16)).

- Select a solution among the best solutions.
- Generate a local solution around the selected best solution.

**end**

- Assign each document to a closest center by calculating the distance to each centroid (clustering).

- Compute the fitness value.

- Generate a new solution by flying randomly.

**if** *rand* <  $A_i$  &  $f(x_i) < f(x^*)$  **then**

- Replace the position with the new one.
- Increase  $r_i$  and reduce  $A_i$  (Equation (17)).

**end**  **end****end****return** *global best position*

## 4.2.3. GWO Algorithm

The GWO algorithm mimics the leadership hierarchy and hunting mechanisms of grey wolves in nature. GWO generates three levels of fitness solutions such as alpha ( $\alpha$ -first), beta ( $\beta$ -second), and delta ( $\delta$ -third). The rest of the candidate solutions are assumed to be omega ( $\omega$ ) [6].

$$\vec{X}(t+1) = \vec{X}_p(t) + \vec{A} \cdot \vec{D}, \vec{D} = \left| \vec{C} \cdot \vec{X}_p(t) - \vec{X}(t) \right| \quad (18)$$

$$\vec{A} = 2\vec{A} \cdot \vec{r}_1 - \vec{a}, \vec{C} = 2\vec{r}_2, \vec{a} = 2 - t \frac{2}{MaxIter} \quad (19)$$

$$\vec{D}_\alpha = \left| \vec{C}_1 \cdot \vec{X}_\alpha - \vec{X} \right|, \vec{D}_\beta = \left| \vec{C}_2 \cdot \vec{X}_\beta - \vec{X} \right|, \vec{D}_\delta = \left| \vec{C}_3 \cdot \vec{X}_\delta - \vec{X} \right| \quad (20)$$

$$\vec{X}_1 = \left| \vec{X}_\alpha - \vec{A}_1 \vec{D}_\alpha \right|, \vec{X}_2 = \left| \vec{X}_\beta - \vec{A}_2 \vec{D}_\beta \right|, \vec{X}_3 = \left| \vec{X}_\delta - \vec{A}_3 \vec{D}_\delta \right| \quad (21)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (22)$$

Here, each GWO position in the GWO algorithm represents a cluster centroid of text document clusters. The initialization phase is used to initialize the number of clusters K and a few GWO parameters as shown in Algorithm 4. After the initialization, Equations (18)–(22) are used to search for new positions. The centroid positions and clusters are created by assigning each document to the closest center (particle's position) by calculating the distance to each centroid. Then, the fitness function evaluates the clusters and the global best (gbest) solution is updated until reaches the stopping criteria. For each iteration, the alpha ( $\alpha$ -first), beta ( $\beta$ -second), and delta ( $\delta$ -third) solutions are updated. As a result, the algorithm finally returns the global best solution [40,41].

To calculate  $\vec{D}$ ,  $\vec{A}$ , and  $\vec{C}$  vectors Equation (18) is used. Equations (20)–(22) are used to update the positions of wolves. Here,  $\vec{a}$  is a control parameter,  $\vec{A}$  and  $\vec{C}$  are coefficient vectors, and  $\vec{r}_1$  and  $\vec{r}_2$  are random vectors.

**Algorithm 4:** GWO document clustering algorithm**Result:** Global best position**Initialization;**

- Randomly choose K cluster centers to initialize a population of n wolf positions.
- Initialize a, A, and C for wolf.
- Calculate the fitness of each search agent.
  - Assign each document to the closest center by calculating the distance to each centroid (clustering).
  - Compute the fitness value.  
 $X_\alpha$  = the best search agent.  
 $X_\beta$  = the second best search agent.  
 $X_\delta$  = the third best search agent.

**while** *Until the stop criteria are met do*  **while** *For each wolf do*

| Update the position of the current search agent (Equation (22)).

**end**

Update a, A, and C.

- Calculate the fitness of each search agent.
  - Assign each document to the closest center by calculating the distance to each centroid (clustering).
  - Compute the fitness value.

  Update  $X_\alpha$ ,  $X_\beta$ , and  $X_\delta$ .**end****return** *global best position*

## 4.2.4. Parameter Set of SI Algorithms

This subsection shows the parameters of all standard SI algorithms and their values. In this study, we focused on comparing the standard SI algorithms. We adopted and used the default parameters of SI algorithms that were introduced in the literature experiments for generic optimization, with different examples [17,42]. Although our experimental problem was text document clustering, which is not an optimization problem, we applied the initial parameter settings of the optimization problem case to ours and tested those standard SI algorithms. We did not use any other parameter fine-tuning methods here.

As shown in Table 1, k represents the number of agents, and its value was 10 in all algorithms for this experiment. In the PSO algorithm parameter values were assigned as  $\omega = 0.9$ ,  $c_1 = 0.5$ , and  $c_2 = 0.3$  [42]. In the BA algorithm parameter values were assigned as  $A = 0.5$ ,  $r = 0.5$ ,  $Q_{min} = 0.0$ , and  $Q_{max} = 2.0$  [17].

**Table 1.** Parameter set for SI algorithms.

PSO	Values	BA	Values	BA	Values
k	10	k	10	k	10
$\omega$	0.9	A	0.5		
C1	0.5	r	0.5		
C2	0.3	Qmin	0.0		
		Qmax	2.0		

## 4.2.5. Features of SI Algorithms

This section shows the features of SI algorithms, the details of important control and local search parameters, and the best performing area of SI algorithms PSO, GWO, and BA as shown in Table 2.

The text document clustering problems are NP problems. Since there is no deterministic algorithm to solve this problem in a polynomial time, we applied an approximate

algorithm to find the nearly optimal solution. The SI algorithms were used to solve the problem in a stochastic manner. Although we could not find the feasibility of the text document clustering in a polynomial time, we could find an approximate feasible solution by using the SI algorithms with 10 simulations for each iteration (1, 10, 20, ..., 100) as mentioned in the experiment section.

In PSO,  $\omega$ ,  $c_1$ , and  $c_2$  are the important control parameters.  $r_1$  and  $r_2$  are the local search parameters, which are drawn from uniform distribution  $[0, 1]$ . PSO is a simple, fast computing, and parallel processing algorithm [43]. Comparing other standard SI algorithms, the standard PSO algorithm performs well in clustering and scheduling applications [14,17,18].

The GWO algorithm includes two main control parameters,  $\vec{a}$  and  $C$ , and local search random vector parameters  $r_1$  and  $r_2$  from  $[0, 1]$ . GWO has a few important features such as faster convergence due to continuous reduction of the search space and fewer decision variables (i.e.,  $\alpha$ ,  $\beta$ , and  $\delta$ ), avoiding local minima, better stability, and robustness [44]. Comparing other standard SI algorithms, GWO performs well in robot swarm learning [13].

The BA has control parameters such as loudness ( $A$ ), pulse rate ( $r$ ), pulse rate adaptation ( $\gamma$ ), loudness decrements ( $\alpha$ ), and local search radius and frequency band parameters ( $Q$ ), and also has local search parameters such as the random vector ( $\beta$ ) from a uniform distribution  $[0, 1]$  and a random walk  $\epsilon$  from the normal distribution  $[-1, 1]$ . The BA provides very quick convergence at the initial stage by switching from exploration to exploitation [45]. Comparing other standard SI algorithms, the BA also performs well in clustering [17].

**Table 2.** Parameters, features, and best performing areas of SI algorithms.

Algorithms	Parameters	Features	Best Performing Area
PSO	Control: $\omega$ , $c_1$ , and $c_2$ ; Local Search: $r_1$ , and $r_2$ are drawn from uniform distribution $[0, 1]$ .	Simple, fast computing speed, and parallel processing [43].	Clustering and scheduling [14,17,18].
GWO	Control: $\vec{a}$ and $C$ ; Local Search: $r_1$ , and $r_2$ are random vectors $[0, 1]$ .	Faster convergence due to continuous reduction of search space and fewer decision variables (i.e., $\alpha$ , $\beta$ , and $\delta$ ); avoiding local minima; better stability and robustness [44].	Robot swarm learning [13].
BA	Control: loudness ( $A$ ), pulse rate ( $r$ ), pulse rate adaptation ( $\gamma$ ), loudness decrements ( $\alpha$ ), local search radius, and frequency band parameters ( $Q$ ); Local Search: $\beta$ is random vector uniform distribution $[0, 1]$ , $\epsilon$ is random walk normal distribution $[-1, 1]$ .	Provides very quick convergence at a very initial stage by switching from exploration to exploitation [45].	Clustering and feature selection [6,17].

## 5. Experiment

The experimentation shows benchmark data sets, experimental conditions, and metrics that were used for the experimentation.

### 5.1. Benchmark Data Sets

In an unsupervised clustering context, classes and class memberships are unknown, and these must be discovered by the clustering algorithm. The performance of these algorithms can be determined by using data sets with known clusters for benchmarking in unsupervised clustering [46]. In this study, we used six different benchmark data sets with different numbers of documents, terms (features), and clusters which were created from BBC Sport news and 20 newsgroup data sets as shown in Table 3. The BBC Sport news data set is from BBC News and is provided to use as a machine learning benchmark consisting of 737 text documents including 4613 terms under five classes: athletics, rugby, cricket, tennis, and football [9]. The 20 newsgroups are popular machine learning data sets for text applications such as text clustering that consist of 20,000 messages taken from 20 different newsgroups such as misc.forsale, soc.religion.christian, and so on [10].

As shown in Table 3, Data set 1 includes 1427 documents and 23,057 terms and these documents belong to 2 clusters from 20 newsgroups. Data set 2 consists of 737 documents and 4613 terms and belongs to 5 clusters from BBC Sport news. Data set 3 includes 40 documents and 2596 terms and belongs to 5 clusters from BBC Sport news. Data set 4 includes 200 documents and 8716 terms and these documents belong to 4 clusters from 20 newsgroups. Data set 5 includes 100 documents and 5549 terms and these documents belong to 3 clusters from 20 newsgroups. Data set 6 includes 100 documents and 3876 terms belonging to 2 clusters from BBC Sport news.

**Table 3.** Benchmark data sets.

Data Set	Source	No. of Documents	No. of Terms	No. of Clusters
1	20 newsgroups	1427	23,057	2
2	BBC Sport	737	4613	5
3	BBC Sport	40	2596	5
4	20 newsgroups	200	8716	4
5	20 newsgroups	100	5549	3
6	BBC Sport	100	3876	2

### 5.2. Experimental Conditions

In this study, Ubuntu OS, Python3, and Jupyter Notebook were used for the experiments. The SI algorithms PSO, BA, and GWO are stochastic. Hence, for the same iteration number different best solutions can be obtained. Therefore, 10 simulations were used for each iteration number, and the best fitness was taken among them for each algorithm.

### 5.3. Results

In this study, we used standard PSO, BA, and GWO for the text document clustering with six data sets, and the performance of these algorithms was evaluated using various metrics such as purity, homogeneity, completeness, V-measure, ARI, and average running time. These results were compared with the K-means clustering algorithm.

Table 4 shows the mean and standard deviation results for all iteration numbers and their performance ranks for all clustering algorithms with six data sets from BBC news and 20 newsgroups. The performance of each algorithm (K-means, PSO, GWO, and BA) was measured using metrics such as purity, homogeneity, completeness, V-measure, and ARI, and those results are displayed in Table 4. The highest purity, homogeneity, completeness, V-measure, and ARI values of each algorithm for the data sets are highlighted in bold. The rank is displayed for each algorithm based on the average evaluation metric values for each and all data sets. Here, the PSO algorithm has the highest purity mean values (0.724, 0.873, 0.689, 0.848, and 0.998) and BA has the lowest purity mean values (0.636, 0.786, 0.607, 0.728, and 0.976) for data sets 1, 3, 4, 5, and 6, respectively. As shown in Table 4, in terms of all metrics, K-means, PSO, GWO, and BA have ranks 3, 1, 2, and 4, respectively, for data sets 1, 3, 4, 5, and 6. K-means achieved the highest purity mean value of 0.929 and BA achieved the lowest purity mean value of 0.692 for data set 2. The other metrics' mean values also provide the same data pattern and rank as purity for all data sets and algorithms as shown in Table 4.

In terms of all evaluation metrics (purity, homogeneity, completeness, V-measure, and ARI), the total performance ranks for algorithms K-means, PSO, GWO, and BA are 3, 1, 2, and 4, respectively. Comparing all results, PSO performs best except for data set 2. Based on the rank, PSO shows the best performance, and the second-best is GWO. The third is K-means and BA performs last.

**Table 4.** Mean and standard deviation results for all iteration numbers for clustering algorithms with six data sets.

Data Sets	Metrics	K-Means	PSO		GWO	BA
			Mean (Std.)			
1	Purity	0.665 (0.04)	<b>0.724 (0.028)</b>		0.695 (0.018)	0.636 (0.02)
	Homogeneity	0.081 (0.041)	<b>0.149 (0.036)</b>		0.108 (0.023)	0.052 (0.014)
	Completeness	0.085 (0.39)	<b>0.153 (0.039)</b>		0.112 (0.023)	0.057 (0.016)
	V-measure	0.083 (0.04)	<b>0.151 (0.037)</b>		0.110 (0.023)	0.054 (0.014)
	ARI	0.112 (0.057)	<b>0.203 (0.045)</b>		0.156 (0.029)	0.0732 (0.019)
Rank		3	1		2	4
2	Purity	<b>0.929 (0.066)</b>	0.775 (0.042)		0.74 (0.02)	0.692 (0.034)
	Homogeneity	<b>0.833 (0.097)</b>	0.525 (0.067)		0.468 (0.041)	0.408 (0.058)
	Completeness	<b>0.854 (0.076)</b>	0.527 (0.068)		0.472 (0.05)	0.413 (0.059)
	V-measure	<b>0.843 (0.088)</b>	0.525 (0.066)		0.470 (0.045)	0.410 (0.058)
	ARI	<b>0.832 (0.118)</b>	0.536 (0.089)		0.462 (0.051)	0.393 (0.0665)
Rank		1	2		3	4
3	Purity	0.734 (0.049)	<b>0.873 (0.025)</b>		0.839 (0.025)	0.786 (0.027)
	Homogeneity	0.591 (0.082)	<b>0.793 (0.059)</b>		0.733 (0.039)	0.637 (0.046)
	Completeness	0.605 (0.082)	<b>0.797 (0.058)</b>		0.735 (0.041)	0.65 (0.046)
	V-measure	0.598 (0.081)	<b>0.795 (0.058)</b>		0.734 (0.04)	0.643 (0.045)
	ARI	0.478 (0.087)	<b>0.727 (0.058)</b>		0.659 (0.067)	0.552 (0.07)
Rank		3	1		2	4
4	Purity	0.592 (0.046)	<b>0.689 (0.04)</b>		0.652 (0.027)	0.607 (0.039)
	Homogeneity	0.271 (0.059)	<b>0.36 (0.056)</b>		0.322 (0.033)	0.284 (0.046)
	Completeness	0.284 (0.062)	<b>0.37 (0.058)</b>		0.336 (0.036)	0.301 (0.051)
	V-measure	0.278 (0.06)	<b>0.365 (0.057)</b>		0.329 (0.034)	0.292 (0.048)
	ARI	0.248 (0.058)	<b>0.363 (0.057)</b>		0.318 (0.029)	0.263 (0.059)
Rank		3	1		2	4
5	Purity	0.709 (0.036)	<b>0.848 (0.04)</b>		0.793 (0.05)	0.728 (0.034)
	Homogeneity	0.309 (0.052)	<b>0.541 (0.098)</b>		0.427 (0.075)	0.314 (0.049)
	Completeness	0.323 (0.063)	<b>0.556 (0.084)</b>		0.448 (0.082)	0.347 (0.076)
	V-measure	0.315 (0.540)	<b>0.548 (0.091)</b>		0.437 (0.078)	0.329 (0.059)
	ARI	0.321 (0.069)	<b>0.598 (0.080)</b>		0.473 (0.117)	0.344 (0.066)
Rank		3	1		2	4
6	Purity	0.994 (0.008)	<b>0.998 (0.006)</b>		0.995 (0.009)	0.976 (0.023)
	Homogeneity	0.958 (0.049)	<b>0.987 (0.041)</b>		0.45 (0.491)	0.864 (0.093)
	Completeness	0.959 (0.048)	<b>0.987 (0.04)</b>		0.972 (0.051)	0.867 (0.089)
	V-measure	0.958 (0.049)	<b>0.987 (0.04)</b>		0.972 (0.051)	0.866 (0.091)
	ARI	0.975 (0.031)	<b>0.992 (0.027)</b>		0.982 (0.035)	0.909 (0.085)
Rank		3	1		2	4
Total rank		3	1		2	4

Figures 2–7 show the comparative results for SI algorithms and K-means with different data sets 1, 2, 3, 4, 5, and 6 using a purity metric.

Figure 2 shows the comparison of the algorithms based on purity with data set 1. This result clearly shows that PSO has the best performance over others and the second best is GWO. K-means and BA have third and fourth positions, respectively.

Figure 3 shows the comparison of the algorithms with data set 2. This result clearly shows that K-means has the best performance over the others and the second best is PSO. GWO and BA have third and fourth positions, respectively. Only for iteration number 1 was less purity observed and for the remaining iterations, stable purity is observed.

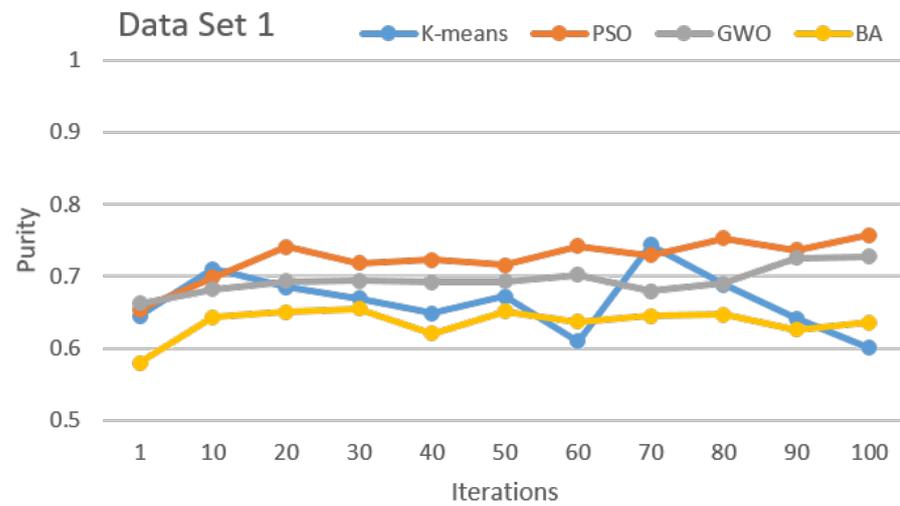


Figure 2. Purity comparison for data set 1.



Figure 3. Purity comparison for data set 2.

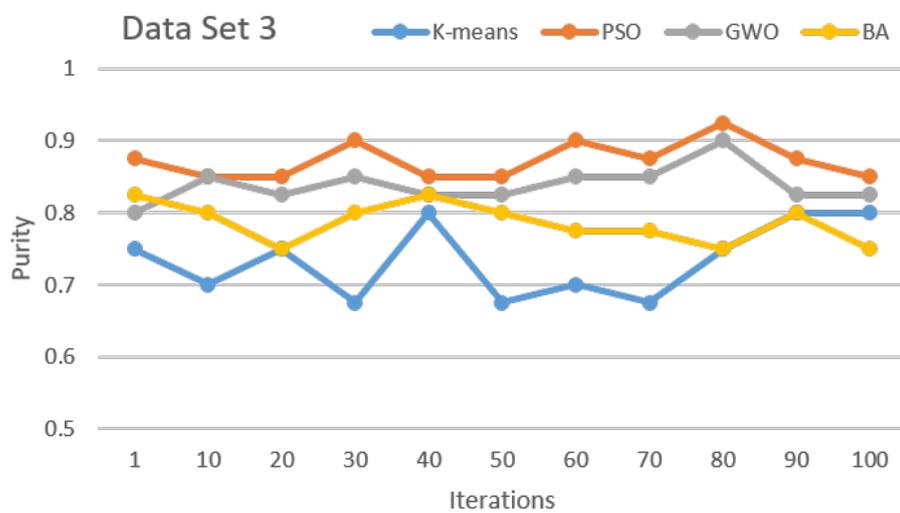


Figure 4. Purity comparison for data set 3.

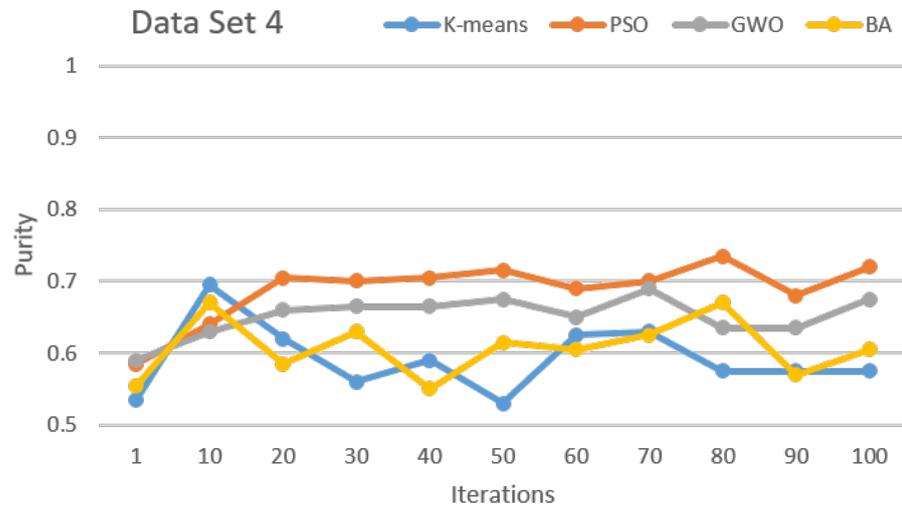


Figure 5. Purity comparison for data set 4.



Figure 6. Purity comparison for data set 5.

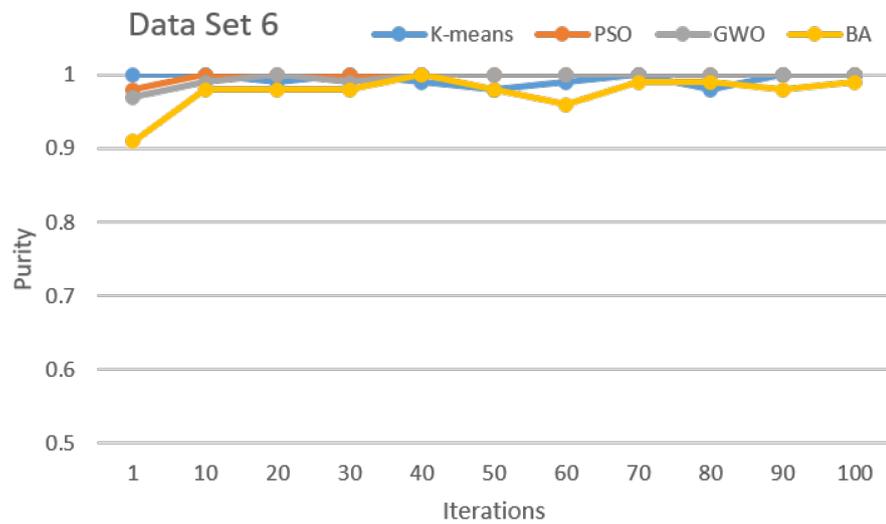


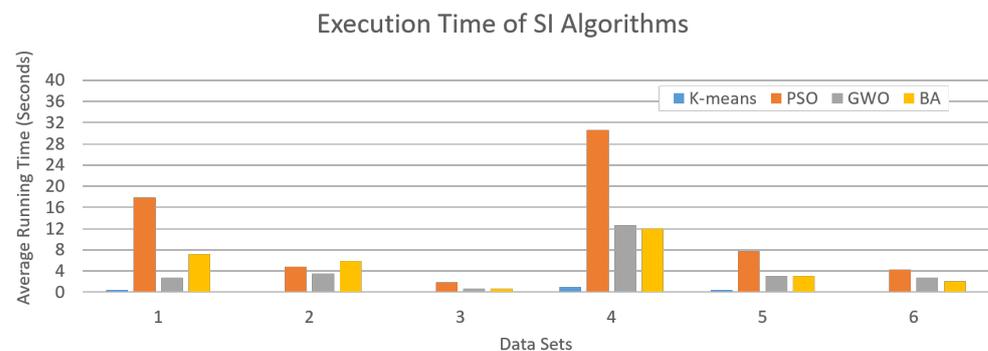
Figure 7. Purity comparison for data set 6.

Figure 4 shows the algorithm comparison with data set 3. It shows the PSO, GWO, and BA perform first, second, and third, respectively. K-means performs lower than all other SI algorithms.

Figures 5 and 6 show a comparison of the algorithms based on data sets 4 and 5, respectively. Here, based on purity results, almost the same results are shown that PSO and GWO perform first and second, respectively. BA and K-means show almost equal and lower performance than PSO and GWO.

Figure 7 shows the algorithm comparison based on data set 6. Here, PSO, GWO, BA, and K-means show almost equal good performance in all iterations.

As shown in Figure 8, the average of the total running time of SI algorithms in each data set was used to evaluate the computational efficiency of the algorithms. Here, K-means is much faster than PSO, GWO, and BA with shorter average running time which is less than a second. Here, PSO takes 17.92, 4.83, 1.83, 30.6, 7.8, and 4.21 average running times in seconds for executing data sets 1, 2, 3, 4, 5, and 6, respectively. This is comparatively more execution time than other algorithms. GWO and BA have an almost equal average running time for all data sets and are faster than the PSO algorithm. Although the PSO has a longer execution time, other evaluation metrics show that the performance of PSO is better at finding the optimum solution.



**Figure 8.** Average running time of SI algorithms.

#### 5.4. Discussion

Based on the above results, on average PSO performs best compared to GWO, BA, and K-means. The GWO algorithm performs well compared to BA and K-means. The results show that almost all algorithms provide stabilized results only after the 20th iteration and above. The results show that the algorithms achieve lower maximum purity (0.75) for a larger number of terms (data sets 1 and 4) and higher maximum purity values (close to 1) for a small number of terms (data sets 2, 3, 5, and 6). However, in terms of computational efficiency, K-means is much faster than other algorithms. Although the average running time of PSO for almost all data sets is relatively (approximately 50%) higher than other algorithms, PSO offers the best performance to find the optimal solution.

Among a number of SI algorithms for solving document clustering, PSO finds the optimal solution compared to other SI and K-means traditional algorithms. The PSO algorithm explores efficiently in search space and exploits the best solutions found by tracking global best and local best solutions. These characteristics help the PSO to solve the document clustering problem efficiently. The GWO algorithm is good at exploring new areas. The document clustering problem is not suitable for the exploring expert. The BA algorithm is good at governing and dominating the prey. The document clustering problem, however, has a whole set of documents together. Applying a velocity vector in the BA does not change the essential grouping of document similarities. That is why the nature of the BA algorithm is not fit for the document clustering problem.

The nature and characteristics of SI algorithms are suitable to apply in many domains such as feature selection, finding an optimal route, job scheduling, role-based learning, and clustering. However, variants of SI algorithms such as improved and hybridized SI

algorithms are performing well in many optimization tasks compared to the standard SI algorithms [6]. Based on proper parameter setting and tuning schemes, the SI algorithms can be applied to the appropriate domain of problem-solving [7].

## 6. Conclusions

The specific characteristics of each SI algorithm are suitable for solving specific optimization problems such as feature selection, finding an optimal route, job scheduling, role-based learning, and clustering. To handle specific optimization tasks, finding the proper SI algorithm is important. In this study, we compared a few important SI algorithms (PSO, BA, and GWO) for the text document clustering problem. These results were also compared with the K-means traditional clustering algorithm. To compare the performance of these algorithms, a few metrics (purity, homogeneity, completeness, V-measure, and ARI) were used with the news articles of six benchmark data sets of various sizes. The comparison of these results shows that the PSO performs best compared to GWO, BA, and K-means. The GWO algorithm performs well compared to the BA and K-means. To compare the computational efficiency, the average running time was used and the results show that compared to other algorithms, K-means takes far less time and PSO takes more time for execution. PSO achieved better results in terms of solving problems but took more time for execution, which demands more study to improve the algorithms in different aspects. As future work, we are aiming to optimize the SI algorithms for better results in the document clustering domain.

**Author Contributions:** Conceptualization, S.S. and E.C.; methodology, S.S. and E.C.; software, S.S.; writing—original draft preparation, S.S.; writing—review and editing, E.C.; project administration, E.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the BK21 FOUR (Fostering Outstanding Universities for Research) funded by the Ministry of Education (MOE, Korea) and National Research Foundation of Korea (NRF).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** This study used data sets from the publicly archived datasets ucd.ie and ucd.edu, and source code from [ju.edu.jo](http://ju.edu.jo) (accessed on 22 April 2021).

**Acknowledgments:** We thank the Global Scholarship Program for Foreign Graduate Students at Kookmin University, Korea, for its support.

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

## References

1. Ramkumar, A.S.; Nethravathy, R. Text Document Clustering using K-means Algorithm. *Int. Res. J. Eng. Technol.* **2019**, *6*, 1164–1168.
2. Chen, F.; Deng, P.; Wan, J.; Zhang, D.; Vasilakos, A.V.; Rong, X. Data mining for the internet of things: Literature review and challenges. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 431047. [[CrossRef](#)]
3. Singh, V.K.; Tiwari, N.; Garg, S. Document Clustering Using K-Means, Heuristic K-Means and Fuzzy C-Means. In Proceedings of the 2011 International Conference on Computational Intelligence and Communication Networks, Gwalior, India, 7–9 October 2011; pp. 297–301. [[CrossRef](#)]
4. Jensi, R.; Wiselin, J. A Survey on Optimization Approaches to Text Document Clustering. *Int. J. Comput. Sci. Appl.* **2013**, *3*, 31–44. [[CrossRef](#)]
5. Hamerly, G.; Drake, J., Accelerating Lloyd's Algorithm for k-Means Clustering. In *Partitional Clustering Algorithms*; Celebi, M.E., Ed.; Springer International Publishing: Cham, Switzerland, 2015; pp. 41–78. [[CrossRef](#)]
6. Hassanien, A.E. *Swarm Intelligence: Principles, Advances, and Applications/About Ella Hassanien, Eid Emary*; CRC Press: Boca Raton, FL, USA, 2016.
7. Selvaraj, S.; Choi, E. Survey of Swarm Intelligence Algorithms. In *ICSIM '20: Proceedings of the 3rd International Conference on Software Engineering and Information Management*; Association for Computing Machinery: New York, NY, USA, 2020; pp. 69–73. [[CrossRef](#)]
8. Brezočnik, L.; Fister, I.; Podgorelec, V. Swarm Intelligence Algorithms for Feature Selection: A Review. *Appl. Sci.* **2018**, *8*, 1521. [[CrossRef](#)]

9. BBC News Datasets. Available online: <http://mlg.ucd.ie/datasets/bbc.html> (accessed on 22 April 2021).
10. 20 Newsgroups. Available online: <https://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html> (accessed on 22 April 2021).
11. Shima Sabet, M.S.; Farokhi, F. A comparison between swarm intelligence algorithms for routing problems. *Electr. Comput. Eng. Int. J. (ECIJ)* **2016**, *5*, 17–33. [[CrossRef](#)]
12. Basir, M.A.; Ahmad, F. Comparison on Swarm Algorithms for Feature Selections/Reductions. *Int. J. Sci. Eng. Res.* **2014**, *5*, 479–486.
13. Fan, J.; Hu, M.; Chu, X.; Yang, D. A comparison analysis of swarm intelligence algorithms for robot swarm learning. In Proceedings of the 2017 Winter Simulation Conference (WSC), Las Vegas, NV, USA, 3–6 December 2017; pp. 3042–3053. [[CrossRef](#)]
14. Mohana, S.J.; Saroja, M.; Venkatachalam, M. Comparative Analysis of Swarm Intelligence Optimization Techniques for Cloud Scheduling. *Int. J. Innov. Sci. Eng. Technol.* **2014**, *1*, 15–19.
15. Elhady, G.F.; Tawfeek, M.A. A comparative study into swarm intelligence algorithms for dynamic tasks scheduling in cloud computing. In Proceedings of the 2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt, 12–14 December 2015; pp. 362–369. [[CrossRef](#)]
16. Zhu, H.; Wang, Y.; Ma, Z.; Li, X. A Comparative Study of Swarm Intelligence Algorithms for UCAV Path-Planning Problems. *Mathematics* **2021**, *9*, 171. [[CrossRef](#)]
17. Gong, X.; Liu, L.; Fong, S.; Xu, Q.; Wen, T.; Liu, Z. Comparative Research of Swarm Intelligence Clustering Algorithms for Analyzing Medical Data. *IEEE Access* **2019**, *7*, 137560–137569. [[CrossRef](#)]
18. Figueiredo, E.; Macedo, M.; Siqueira, H.V.; Santana, C.J.; Gokhale, A.; Bastos-Filho, C.J. Swarm intelligence for clustering—A systematic review with new perspectives on data mining. *Eng. Appl. Artif. Intell.* **2019**, *82*, 313–329. [[CrossRef](#)]
19. Yeoh, J.M.; Caraffini, F.; Homapour, E.; Santucci, V.; Milani, A. A Clustering System for Dynamic Data Streams Based on Metaheuristic Optimisation. *Mathematics* **2019**, *7*, 1229. [[CrossRef](#)]
20. Florez-Lozano, J.M.; Caraffini, F.; Parra, C.; Gongora, M. A Robust Decision-Making Framework Based on Collaborative Agents. *IEEE Access* **2020**, *8*, 150974–150988. [[CrossRef](#)]
21. Florez-Lozano, J.; Caraffini, F.; Parra, C.; Gongora, M. Cooperative and distributed decision-making in a multi-agent perception system for improvised land mines detection. *Inf. Fusion* **2020**, *64*, 32–49. [[CrossRef](#)]
22. Lu, Y.; Wang, S.; Li, S.; Zhou, C. Text Clustering via Particle Swarm Optimization. In Proceedings of the 2009 IEEE Swarm Intelligence Symposium, Nashville, TN, USA, 30 March–2 April 2009; pp. 45–51. [[CrossRef](#)]
23. Judith, J.; Jayakumari, J. Distributed document clustering analysis based on a hybrid method. *China Commun.* **2017**, *14*, 131–142. [[CrossRef](#)]
24. Abualigah, L.; Gandomi, A.H.; Elaziz, M.A.; Hussien, A.G.; Khasawneh, A.M.; Alshinwan, M.; Houssein, E.H. Nature-Inspired Optimization Algorithms for Text Document Clustering—A Comprehensive Analysis. *Algorithms* **2020**, *13*, 345. [[CrossRef](#)]
25. Rashaideh, H.; Sawaie, A.; Al-Betar, M.A.; Abualigah, L.M.; Al-laham, M.M.; Al-Khatib, R.M.; Braik, M. A Grey Wolf Optimizer for Text Document Clustering. *J. Intell. Syst.* **2020**, *29*, 814–830. [[CrossRef](#)]
26. Abualigah, L.M. Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. *J. Supercomput.* **2017**, *73*, 4773–4795. [[CrossRef](#)]
27. Suganya Selvaraj, H.K.; Choi, E. Offline-to-Online Service and Big Data Analysis for End-to-end Freight Management System. *J. Inf. Process. Syst.* **2020**, *16*, 377–393.
28. Abualigah, L.M.; Khader, A.T.; Hanandeh, E.S. A new feature selection method to improve the document clustering using particle swarm optimization algorithm. *J. Comput. Sci.* **2018**, *25*, 456–466. [[CrossRef](#)]
29. Sailaja, D.; Kishore, M.; Jyothi, B.; Prasad, N. An Overview of Pre-Processing Text Clustering Methods. *Int. J. Comput. Sci. Inform. Technol.* **2015**, *6*, 3119–3124.
30. Tan, Y. Chapter 1—Introduction. In *Gpu-Based Parallel Implementation of Swarm Intelligence Algorithms*; Tan, Y., Ed.; Morgan Kaufmann: San Francisco, CA, USA, 2016; pp. 1–7. [[CrossRef](#)]
31. Karol, S.; Mangat, V. Evaluation of text document clustering approach based on particle swarm optimization. *Open Comput. Sci.* **2013**, *3*, 69–90. [[CrossRef](#)]
32. Purity Metric. Available online: <http://www.cse.chalmers.se/~richajo/dit862/L13/Text%20clustering.html> (accessed on 22 April 2021).
33. Rosenberg, A.; Hirschberg, J. V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In *EMNLP-CoNLL '12: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*; Association for Computational Linguistics: Prague, Czech Republic, 2007; pp. 410–420.
34. Sklearn Metrics. Available online: <https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics> (accessed on 22 April 2021).
35. Steinley, D. Properties of the Hubert-Arable Adjusted Rand Index. *Psychol. Methods* **2004**, *9*, 386–396. [[CrossRef](#)] [[PubMed](#)]
36. Cui, X.; Potok, T.E.; Palathingal, P. Document clustering using particle swarm optimization. In Proceedings the 2005 IEEE Swarm Intelligence Symposium, SIS 2005, Pasadena, CA, USA, 8–10 June 2005; pp. 185–191. [[CrossRef](#)]
37. PSO Code. Available online: <https://github.com/dandynaufaldi/particle-swarm-optimized-clustering> (accessed on 22 April 2021).
38. Wang, G.; Chang, B.; Zhang, Z. A multi-swarm bat algorithm for global optimization. In Proceedings the 2015 IEEE Congress on Evolutionary Computation (CEC), Sendai, Japan, 25–28 May 2015; pp. 480–485. [[CrossRef](#)]

39. BA Code. Available online: <https://github.com/herukurniawan/bat-algorithm/blob/master/BatAlgorithm.py> (accessed on 22 April 2021).
40. GWO Code 1. Available online: <https://github.com/7ossam81/EvoloPy> (accessed on 22 April 2021).
41. GWO Code 2. Available online: [https://github.com/czeslavo/gwo/blob/master/optimization/grey\\_wolf\\_optimizer.cpp](https://github.com/czeslavo/gwo/blob/master/optimization/grey_wolf_optimizer.cpp) (accessed on 22 April 2021).
42. PySwarms. Available online: <https://pyswarms.readthedocs.io/en/latest/api/pyswarms.single.html> (accessed on 22 April 2021).
43. Li, M.; Du, W.; Nian, F. An Adaptive Particle Swarm Optimization Algorithm Based on Directed Weighted Complex Network. *Math. Probl. Eng.* **2014**, *2014*, 434972. [[CrossRef](#)]
44. Hameed, I.A.; Bye, R.T.; Osen, O.L. Grey wolf optimizer (GWO) for automated offshore crane design. In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 6–9 December 2016; pp. 1–6. [[CrossRef](#)]
45. Yang, X.; He, X. Bat algorithm: literature review and applications. *arXiv* **2013**, arXiv:1308.3900.
46. Mechelen, I.; Boulesteix, A.; Dangl, R.; Dean, N.; Guyon, I.; Hennig, C.; Leisch, F.; Steinley, D. Benchmarking in cluster analysis: A white paper. *arXiv* **2018**, arXiv:1809.10496.