

# Text documents clustering using modified multi-verse optimizer

Ammar Kamal Abasi<sup>1</sup>, Ahamad Tajudin Khader<sup>2</sup>, Mohammed Azmi Al-Betar<sup>3</sup>, Syibrah Naim<sup>4</sup>,  
Mohammed A. Awadallah<sup>5</sup>, Osama Ahmad Alomari<sup>6</sup>

<sup>1,2</sup>School of Computer Sciences, Universiti Sains Malaysia (USM), Malaysia

<sup>3</sup>Department of information technology, Al-Huson University College, Jordan

<sup>4</sup>Technology Department, Endicott College of International Studies (ECIS), Woosong University, Korea

<sup>5</sup>Department of Computer Science, Al-Aqsa University, Palestine

<sup>6</sup>Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Turkey

## Article Info

### Article history:

Received Mar 29, 2020

Revised May 2, 2020

Accepted May 18, 2020

### Keywords:

Multi-verse optimizer

Optimization

Swarm intelligence

Test document clustering

## ABSTRACT

In this study, a multi-verse optimizer (MVO) is utilised for the text document clustering (TDC) problem. TDC is treated as a discrete optimization problem, and an objective function based on the Euclidean distance is applied as similarity measure. TDC is tackled by the division of the documents into clusters; documents belonging to the same cluster are similar, whereas those belonging to different clusters are dissimilar. MVO, which is a recent metaheuristic optimization algorithm established for continuous optimization problems, can intelligently navigate different areas in the search space and search deeply in each area using a particular learning mechanism. The proposed algorithm is called MVOTDC, and it adopts the convergence behaviour of MVO operators to deal with discrete, rather than continuous, optimization problems. For evaluating MVOTDC, a comprehensive comparative study is conducted on six text document datasets with various numbers of documents and clusters. The quality of the final results is assessed using precision, recall, F-measure, entropy accuracy, and purity measures. Experimental results reveal that the proposed method performs competitively in comparison with state-of-the-art algorithms. Statistical analysis is also conducted and shows that MVOTDC can produce significant results in comparison with three well-established methods.

Copyright © 2020 Institute of Advanced Engineering and Science.  
All rights reserved.

## Corresponding Author:

Ammar Kamal Abasi,  
School of Computer Sciences,  
Universiti Sains Malaysia (USM),  
11800 Pulau Pinang, Malaysia.  
Email: ammar\_abasi@student.usm.my

## 1. INTRODUCTION

In the current digital era, massive amounts of online text documents inundate the web every day. Manipulating these text documents is important for improving the query results returned by search engines, unsupervised text organisation systems, text classification, text summarization, knowledge extraction processes, information retrieval services and text mining processes, scientific document clustering [1]. Many approaches have been proposed for unsupervised organising text documents.

Text document clustering (TDC) is an effective and efficient technique used by researchers in this domain [2], this field of text mining enables the organisation of large amounts of textual data. It can be defined as an unsupervised automatic document clustering technique that utilises the document similarities rule to divide documents into homogeneous clusters. In other words, text documents in the same cluster are similar,

whereas those in different clusters are dissimilar [3]. Conventionally, clustering methods can be classified into two main groups: (i) partitional clustering and (ii) hierarchical clustering.

K-means and K-medoids are simple and easy-to-use methods that can be tailored to suit large-scale text document datasets. They are iterative clustering-based techniques initiated with predefined numbers of cluster centroids. At each iteration, documents are distributed into clusters according to similarity functions, depending on the distance between each centroid and its closest document. Then, the cluster centroid is iteratively updated according to the documents belonging to the same cluster. This operation is stopped when all the documents are moved into the right cluster by means of stagnated cluster centroids. The main shortcoming of these methods is their convergence behaviour; they move in one direction in a single search space region and do not perform a wider scan in the whole search space region. Therefore, they can easily become trapped in local optima due to the unknown shapes of search spaces. Given that K-means is a local search area method and TDC is formulated as an optimization problem [4], optimization methods that can escape the local optima can be utilized for TDC [5].

The most successful algorithms recently utilized for TDC are metaheuristic-based algorithms [6]. The first type of metaheuristic algorithms is evolutionary-based algorithms (EA), which is initiated with a group of provisional individuals called population. Generation after generation, the population is evolved on the basis of three main operators: *recombination* for mixing the individual features, *mutation* for diversifying the search and selection for utilising the survival-of-the-fittest principle. The EA is stopped when no further evolution can be achieved. The main shortcoming of EAs is that although they can simultaneously navigate several areas in the search space, they cannot perform deep searching in each area to which they navigate. Consequently, EAs mostly suffer from premature convergence. EAs that have been successfully utilized for TDC include genetic algorithm (GA) [7] and harmony search [8].

The second type of metaheuristic algorithms is trajectory-based algorithms (TA); a single solution is used to launch such an algorithm. This solution is improved by repetition using neighbouring-moves operators until a local optimal solution that is in the same search space region, is reached [9]. While TAs can extensively search the initial solution search region and achieve local optima, they cannot navigate simultaneously numerous search space regions. The main TAs utilized for TDC are K-means and K-medoids. Other TAs used for TDC are self-organizing maps (som) [10], and  $\beta$ -hill climbing.

The last type of metaheuristic algorithms is swarm intelligence (SI); an SI algorithm is also initiated with a set of random solutions called a swarm. Iteration after iteration, the solutions in the swarm are reconstructed by means of attracting them by the best solutions that are so far found [11]. SI-based algorithms can easily converge prematurely. Several SI-based TDC are utilized, such as particle swarm optimization (PSO) [12] and artificial bee colony [13].

The multi-verse optimizer (MVO) algorithm was recently proposed as a stochastic population-based algorithm [14] inspired by multi-verse theory. The big bang theory explains the origin of the universe to have been a massive explosion. According to this theory, the origin of everything in our universe requires one big bang. Multi-verse theory believes that more than one explosion (big bang) occurred, with each big bang creating a new and independent universe. This theory is modelled as an optimization algorithm with three concepts: white hole, black hole and wormhole, for performing exploration, exploitation and local search, respectively. The MVO has been utilized for a wide range of optimization problems, such as identifying the optimal parameters of PEMFC stacks [15], unmanned aerial vehicle path planning [16], clustering problems [17], feature selection [18], neural networks [19], and optimising SVM parameters [18].

This paper adapts the MVO algorithm for the TDC problem using Euclidean distance as similarity measure. The adaptation includes modifying the convergence behaviour of MVO operators to deal with the discrete, rather than continuous, optimization problem. The main advantage of the proposed method is that it improves the quality of final outcomes for TDC problems. A comprehensive comparative study is conducted on six text document benchmark datasets that have different numbers of clusters and documents. The quality of the final results is analysed with a discussion using accuracy, precision, recall, F-measure, purity and entropy criteria. The findings of the experimental analyses reveal that the proposed method performs competitively in comparison with state-of-the-art algorithms.

The rest of this paper is organised as follows. Section 2. presents the structure of MVO for TDC. Section 3. discusses the experiments results of MVO. Section 4. gives the conclusion of this study and future work for the authors.

## 2. MULTI-VERSE OPTIMIZER FOR TEXT DOCUMENT CLUSTERING

This section describes the main components utilized for tackling TDC. The term ‘components’ is used to denote the adaptation elements that are conducted for solving the TDC problem using the MVO algorithm and their sequence, including (i) TDC pre-processing, (ii) solution representation and (iii) calculation of the objective function (evaluation of the solutions). Finally, the question of how MVO is adapted to TDC is addressed.

### 2.1. Text document clustering (TDC)

TDC aims to divide documents into clusters; each cluster has similar documents, whereas documents in different clusters are dissimilar [20, 21].

In the use of any clustering algorithm for TDC, the text needs some necessary and preliminary steps (pre-processing); this step filters unnecessary data, such as special formatting, special characters and numbers, out of the text. Thereafter, the pre-processed text document terms are converted into numerical form for further processing. The main goal of this step is to improve the quality of features and reduce the implementation complexity of the TDC algorithm [22]. The text pre-processing includes tokenization, stop word removal and stemming, which are discussed in detail in the succeeding subsections.

#### 2.1.1. Tokenization Phase

In the tokenization phase, each document is broken down into a set of tokens (words), where the token is any sequence of characters separated by spaces. Each document is then formulated as a word instance count (pieces) as a bag-of-words model [23]. Note that the word instance count is filtered through removal of empty sequences, number formatting and collapsing, among other tasks [24].

#### 2.1.2. Stop word removal Phase

In the stop words removal phase, commonly repeated terms (e.g., ‘a’, ‘an’, ‘the’, ‘who’, ‘be’, ‘about’, ‘again’, ‘any’, ‘against’), pronouns (e.g. ‘she’, ‘he’, ‘it’), conjunctions (e.g. ‘but’, ‘and’, ‘or’, ) and similar words are removed because they have high frequencies and negatively affect the clustering process (i.e. they hinder the clustering algorithm.). This process improves the clustering performance and reduces the number of processed words or terms [22].

#### 2.1.3. Stemming Phase

Stemming is the process of decomposing terms to their roots by removal of affixes (prefixes and suffixes) [25]. For example, the root of the word ‘stemming’ is ‘stem’. In the English language, many terms may share the same root; for example, the words ‘connects’, ‘connected’ and ‘connecting’ all stem from the same root, which is ‘connect’ in [www.text-processing.com/demo/stem/](http://www.text-processing.com/demo/stem/). The stemming process attempts to improve the clustering by reducing the number of different terms that have similar grammatical properties and stem from a single term [26].

#### 2.1.4. Solution representation

Each solution is represented as a vector  $\mathbf{x} = (x_1, x_2, \dots, x_d)$ , where  $d$  is the number of documents. Figure 1 shows an example of a solution representation. In this example, five clusters contain twenty document; for example, cluster two has three documents {4, 8, 9}, and document 10 (i.e.,  $x_{10}$ ) is in cluster three.



Figure 1. Solution representation

#### 2.1.5. Objective function

In this study, for each solution  $\mathbf{x}$  the objective function is calculated using the average distance of documents to the cluster centroid (ADDC) as shown in (1).

$$\min f(\mathbf{x}) = \frac{\sum_{i=1}^k (\frac{1}{n_i} \sum_{j=1}^{n_i} D(C_{ki}, d_j))}{k} \quad (1)$$

where  $D(C_{ki}, d_j)$  is the distance between the cluster centroid  $j$  and document  $i$ ,  $n_i$  is the number of documents in cluster  $i$ ,  $k$  is the number of clusters, and  $f(x)$  is the objective function (i.e. minimize the distance).

**2.2. Multi-verse in optimization context**

MVO [14] inspired by multi-verse theory. According to multi-verse theory, universes connect and might even collide with each other. MVO engages and reformulates using three main concepts: wormholes, black holes, and white holes. The probability is used for determining the inflation rate (corresponds to the objective function in the optimization context) for the universe, thereby allowing the universe to assign one of these holes. Given that the universe has a high inflation rate, the probability of a white hole existing increases. Meanwhile, a low inflation rate leads to increased probability of a black hole existing [14]. Regardless of the universe’s inflation rate, wormholes move objects towards the best universe randomly [15, 27]. The black and white hole concepts in MVO are formulated for exploring search spaces, and the wormhole concept is formulated for exploiting search spaces. In other EAs, MVO is initiated by a population of individuals (universes). Thereafter, MVO improves these solutions until a stopping criterion. The conceptual model of the MVO in [14] shows the movements of the objects between the universes via white/black hole tunnels. These hole tunnels are created between two universes on the basis of the inflation rate of each universe (i.e. one universe has a higher inflation rate than the other universes.). Objects move from universes with high inflation rates using white holes. These objects are received by universes with low inflation rates using black holes.

After a population of solutions is initiated, all solutions in MVO are sorted from high inflation rates to low ones. Thereafter, it visits the solutions one by one to attract these solution to the best one. This is done under the assumption that the solution that has been visited has the black hole. As for the white holes, the roulette wheel mechanism is used for selecting one solution.

**2.3. Adapting MVO for TDC**

After the pre-processing step, MVO is used to split documents into their parent clusters. In this study, the solution representation and the objective function formulated above are used. The steps of classical MVO in [14] are adopted for TDC with certain modifications. These modifications are related to the nature of the problem variables. Given that the clustering problem is discrete in nature [28] and MVO was originally proposed for continuous optimization problems, MVO should deal with discrete values of the decision variables of each TDC solution. During the MVO execution, the generation function and the wormhole equations (2) is adjusted for deciding the feasible solution as follows:

$$x_i^j = \begin{cases} \begin{cases} x_j + TDR \times ((ub_j - lb_j) \times r_4 + lb_j)Bigr & r_3 < 0.5, \\ x_j - TDR \times ((ub_j - lb_j) \times r_4 + lb_j)Bigr & r_3 \geq 0.5 \end{cases} & r_2 < WEP \\ x_i^j & r_2 \geq WEP \end{cases} \quad (2)$$

A general overview of MVO for TDC is provided via Figure 2, which visualises the procedural steps.

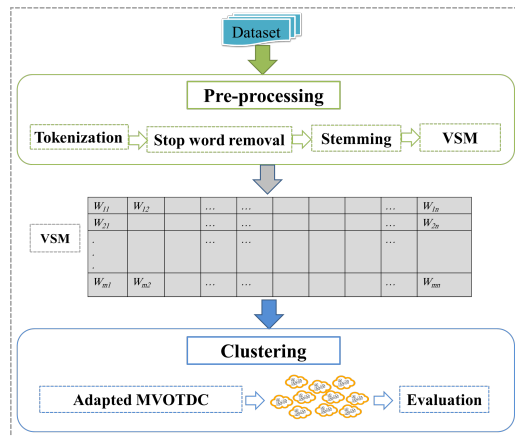


Figure 2. Process of MVOTDC

### 3. EXPERIMENTAL RESULTS

For evaluating the performance of the proposed method, a set of designed experiments is conducted using six instances of standard datasets formulated for measuring the performance of text clustering techniques. Six evaluation measures are used, as conventionally done: precision, recall, F-measure, entropy accuracy, and purity criteria. For comparative evaluation, results obtained in terms of evaluation measures are compared with those obtained by three state-of-the-art algorithms (K-means clustering, GA and PSO) using the same objective function. The experiments are conducted using the programming language MATLAB. Thorough descriptions about the experimental results are given in the following subsections.

#### 3.1. Standard datasets

Table 1 provides the characteristics of the six text document datasets used in this study: (CSTR2, 20Newsgroups, Classic4) in sites.labic.icmc.usp.br/text-collections, (tr12, tr41 and Wap) in glaros.dtc.umn.edu/gkhome/fetch/sw/cluto/datasets.tar.gz.

Table 1. Text document dataset characteristics

ID	Datasets	Number of documents (d)	Number of features or terms (t)	Number of clusters (K)
DS1	CSTR	299	1725	4
DS2	20Newsgroups	300	2275	3
DS3	tr12	313	5329	8
DS4	tr41	878	6743	10
DS5	Wap	1560	7512	20
DS6	Classic4	2000	6500	4

#### 3.2. Results and discussion

The results obtained by MVOTDC are summarised in Table 2, and the parameter settings used in the experiments are given in Table 3. The results are summarised in terms of precision, recall, F-measure, entropy accuracy, and purity for the six datasets. The findings prove the validity and effectiveness of the proposed MVOTDC in the distribution of text documents to the right clusters.

The results are also conducted to show the validity of the proposed method in comparison with three well-known methods: GA, K-means and PSO. Table 3 shows the parameter setting values for each compared algorithm. These parameter settings are used as suggested in [17].

A comparative analysis of K-means, GA, PSO and MVOTDC is provided in Table 2 in terms of precision, recall, F-measure, entropy accuracy, and purity; the average values for each measure are recorded. The results obtained by the K-means clustering algorithm are worse than those obtained by the other algorithms for nearly all datasets. The possible justification is that K-means is a local search algorithm; therefore, it is highly likely to fall in local optima due to its inability to explore the problem search space effectively. Meanwhile, population-based metaheuristic algorithms, such as GA, PSO and MVOTDC, can explore different areas in the search space simultaneously and can consequently achieve better exploration properties.

Table 2 also show that MVOTDC attains minimum entropy and maximum purity, precision, recall, F-measure, accuracy for five datasets (i.e. DS1, DS2, DS3, DS4, DS6). This ability of the proposed MVOTDC algorithm during the search in reaching the right balance between exploitation and exploration with a powerful learning mechanism strengthens its performance in achieving impressive outcomes in comparison with the other methods.

Table 2 provides the results of the F-measure for all compared methods, including MVOTDC. Notably, MVOTDC produces the best F-measure values for five datasets. Furthermore, GA, PSO and MVOTDC outperform K-means in all the datasets.

From a different perspective, Table 2 also shows the accuracy of all compared algorithms. In general, the results obtained by MVOTDC are better than those of the other methods. In fact, the results could be slightly changed from one dataset to another due to the fact that clustering algorithms are normally highly sensitive to the dataset search space. This is can be validated by the finding that MVOTDC obtains the best accuracy in five datasets and the second-best for DS5.

The purity measure of clusters is another external evaluation. It measures the maximum class for each cluster. In general, the closer the purity value to 1, the better the clustering solution. Table 2 shows the results of the purity measure for all compared methods on all datasets. MVOTDC outperforms K-means, GA and PSO in five datasets (i.e. DS1, DS2, DS3, DS4, DS6). The proposed algorithm obtains a 21.5% improvement percentage for DS1 in accordance with K-means. For DS2, MVOTDC's purity values show improvements of

6.0%, 2.6% and 2.4% over those acquired by K-means, GA and PSO, respectively. Meanwhile, the obtained improvements are 15.4%, 9.3%, 5.7%, 19.7%, 4.7%, 2.9%, 8.0%, 4.2% and 4.9% for text document standard datasets DS3, DS4 and DS6. In summary, the results shown in Table 2 reveal that MVOTDC outperforms all compared algorithms in terms of cluster quality (i.e. F-measure and purity).

Entropy is another external measure used in evaluating and comparing the quality of clustering algorithms. The entropy value is zero only when all documents in a single class are placed in a single cluster. In this case, the one cluster solution is considered the best. Table 2 shows the entropy measure values obtained by all the compared algorithms on the different datasets. The bigger the entropy value, the worse the clustering solution. According to the results, MVOTDC provides low entropy values for most of the datasets, which means that it performs better than the other algorithms and offers the best clustering solution. Notably, K-means produces the worst entropy measure for all datasets, whereas GA and PSO are again ranked in between MVOTDC and K-means. The superior performance of MVOTDC is due to its explorative capability in the search space.

The objective function is determined by ADDC for all clustering algorithms so that the distance between the documents in each cluster is minimized. Figure 3 depicts the convergence trends of GA, PSO and MVOTDC using ADDC values. The x-axis is the stream of iteration numbers, whereas the y-axis is the stream of ADDC values. Notably, the convergence rate of MVOTDC is fairly fast for all datasets except DS5.

Table 2. Results of the Accuracy, Precision, Recall, F-measure, Purity and Entropy for K-means, GA, PSO, and MVOTDC algorithms over 30 independent runs for DS1, DS2, DS3, DS4, D65 ,and DS6

Dataset	Measure	Optimization algorithms and techniques			
		K-means [17]	GA [29]	PSO [12]	MVOTDC
DS1	Accuracy	0.3573	0.3398	0.4355	<b>0.4593</b>
	Precision	0.4091	0.4416	0.5340	<b>0.571</b>
	Recall	0.3091	0.3417	0.4359	<b>0.4829</b>
	F-Measure	0.3459	0.3886	0.4819	<b>0.5243</b>
	Purity	0.3524	0.4050	0.4953	<b>0.5684</b>
	Entropy	0.8201	0.7170	0.6199	<b>0.5206</b>
DS2	Accuracy	0.3180	0.3675	0.3498	<b>0.4044</b>
	Precision	0.3121	0.4209	0.4134	<b>0.4391</b>
	Recall	0.3099	0.3676	0.3496	<b>0.384</b>
	F-Measure	0.3406	0.3935	0.3803	<b>0.4109</b>
	Purity	0.3741	0.4080	0.4096	<b>0.4343</b>
	Entropy	0.8028	0.7546	0.7722	<b>0.7120</b>
DS3	Accuracy	0.2971	0.3676	0.4075	<b>0.4485</b>
	Precision	0.3522	0.4128	0.4297	<b>0.5075</b>
	Recall	0.2944	0.3549	0.4263	<b>0.4398</b>
	F-Measure	0.3221	0.3826	0.4277	<b>0.4705</b>
	Purity	0.3907	0.4512	0.4877	<b>0.5448</b>
	Entropy	0.7137	0.6233	0.5719	<b>0.5224</b>
DS4	Accuracy	0.4125	0.4320	<b>0.4870</b>	0.4630
	Precision	0.3944	0.4140	0.4505	<b>0.4568</b>
	Recall	0.3812	0.4007	<b>0.4496</b>	0.4418
	F-Measure	0.3876	0.4071	0.4497	<b>0.4568</b>
	Purity	0.4107	0.5602	0.5789	<b>0.6081</b>
	Entropy	0.5874	0.5469	0.5391	<b>0.5355</b>
DS5	Accuracy	0.5011	0.5316	<b>0.5622</b>	0.5291
	Precision	0.4626	0.5313	<b>0.5249</b>	0.5213
	Recall	0.4010	0.4705	<b>0.4810</b>	0.4496
	F-Measure	0.4314	0.4997	<b>0.5016</b>	0.4831
	Purity	0.4759	0.4916	<b>0.6124</b>	0.6069
	Entropy	0.7043	0.6216	<b>0.5765</b>	0.6625
DS6	Accuracy	0.5858	0.6620	0.6363	<b>0.7042</b>
	Precision	0.5698	0.6725	0.6603	<b>0.6919</b>
	Recall	0.5259	0.6319	0.6163	<b>0.6843</b>
	F-Measure	0.5471	0.6518	0.6377	<b>0.6880</b>
	Purity	0.5938	0.6319	0.6242	<b>0.6742</b>
	Entropy	0.5600	0.5780	0.5306	<b>0.5112</b>

Table 3. Parametric values for different variants of TDC algorithms

Algorithm	Parameters	Value
All Optimization algorithms	Population size	60
All Optimization algorithms	Maximum number of iteration runs	1000
proposed method (MVOTDC)	WEP Max	1
proposed method (MVOTDC)	WEP Min	0.2
proposed method (MVOTDC)	p	6
GA	Crossover probability	0.80
GA	Mutation probability	0.02
PSO	Maximum inertia weight	0.9
PSO	Minimum inertia weight	0.2
PSO	C1	2
PSO	C2	2

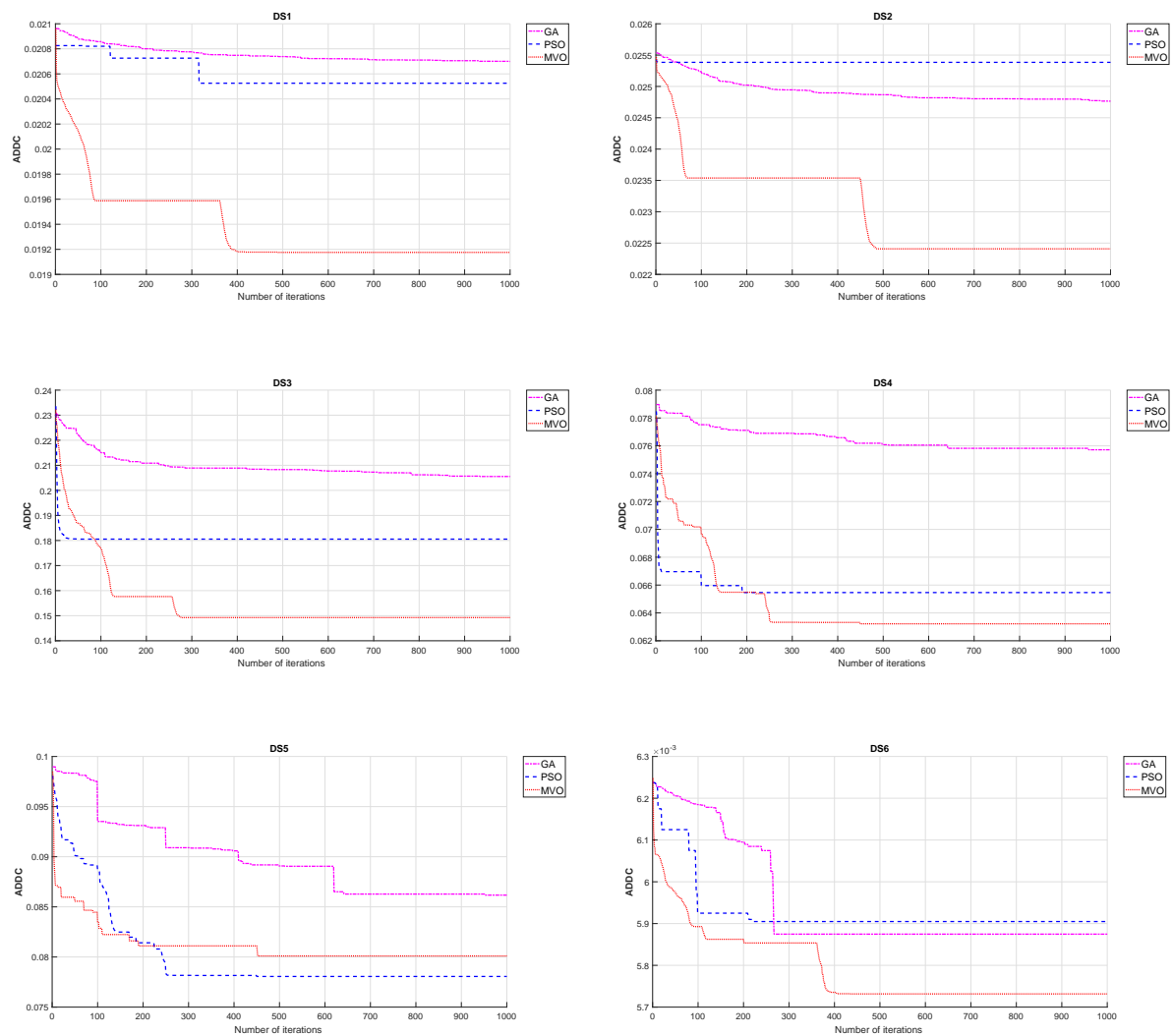


Figure 3. Convergence characteristics of GA, PSO and MVOTDC on datasets D1, D2, D3, D4, D5 and D6

It is worth emphasizing the MVOTDC can be used to address specific optimization problems such as EEG signals denoising [30], gene selection problem [31], and power scheduling problems [32]. Despite the MVOTDC’s superiority among the competitive algorithms, MVOTDC remains sensitive to the characteristics of the datasets, making it difficult to predict its behavior on new datasets while implemented.

#### 4. CONCLUSION AND FUTURE WORK

This paper proposes a metaheuristic optimization algorithm called multi-verse optimizer (MVO) for solving the text document clustering (TDC) problem, i.e. MVOTDC. This method introduces a new strategy of sharing information between solutions on the basis of an objective function and learns from the best solution instead of the global best (i.e. all solutions). The convergence of the results of MVOTDC is impressive due to the method's achievement of the appropriate balance between exploitation and exploration search during each run.

The proposed MVOTDC is evaluated using six text document datasets with various sizes and complexities. The numbers of documents and clusters in each dataset are given. The quality of the obtained results is assessed using six measures: precision, recall, F-measure, entropy accuracy, and purity.

These measures are also used for a comparative evaluation in which three well-known clustering algorithms are used: K-means, genetic algorithm (GA) and particle swarm optimisation (PSO). For all measures, the results obtained by MVOTDC are significantly better than those produced by the three compared methods. In terms of computational time, MVOTDC is slower than K-means and requires nearly the same computational time as GA and PSO. Therefore, MVOTDC can be considered an efficient clustering method for the text clustering domain.

Given the successful outcomes of MVO for the TDC problem, MVOTDC can be implemented for different types of clustering problems. MVO can also be further improved by the addition or modification of its operators so that it can address other discrete optimisation problems, such as scheduling. In addition, datasets other than those used in this work can be used in future studies. In addition, hybridized the MVO with local search strategies in order to improve initial solutions and the exploitation capability during optimization process.

#### REFERENCES

- [1] N. Saini, S. Saha, and P. Bhattacharyya, "Automatic scientific document clustering using self-organized multi-objective differential evolution," *Cognitive Computation*, pp. 1–23, 2018.
- [2] I. Arin, M. K. Erpam, and Y. Saygı, "I-twec: Interactive clustering tool for twitter," *Expert Systems with Applications*, vol. 96, pp. 1–13, 2018.
- [3] A. K. Abasi, A. T. Khader, M. A. Al-Betar, S. Naim, S. N. Makhadmeh, and Z. A. A. Alyasseri, "Link-based multi-verse optimizer for text documents clustering," *Applied Soft Computing*, vol. 87, 2020.
- [4] W. Song, W. Ma, and Y. Qiao, "Particle swarm optimization algorithm with environmental factors for clustering analysis," *Soft Computing*, vol. 21, no. 2, pp. 283–293, 2017.
- [5] Z. A. A. Alyasseri, A. T. Khader, M. A. Al-Betar, M. A. Awadallah, and X.-S. Yang, "Variants of the flower pollination algorithm: a review," *Nature-Inspired Algorithms and Applied Optimization*, pp. 91–118, 2018.
- [6] A. K. Abasi, A. T. Khader, M. A. Al-Betar, S. Naim, S. N. Makhadmeh, and Z. A. A. Alyasseri, "A text feature selection technique based on binary multi-verse optimizer for text clustering," *IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology*, pp. 1–6, 2019.
- [7] J.-H. Jiang, J.-H. Wang, X. Chu, and R.-Q. Yu, "Clustering data using a modified integer genetic algorithm (iga)," *Analytica Chimica Acta*, vol. 354, no. 1, pp. 263–274, 1997.
- [8] R. Forsati, M. Mahdavi, M. Shamsfard, and M. R. Meybodi, "Efficient stochastic algorithms for document clustering," *Information Sciences*, vol. 220, pp. 269–291, 2013.
- [9] O. A. Alomari, A. T. Khader, M. A. Al-Betar, and M. A. Awadallah, "A novel gene selection method using modified mrmr and hybrid bat-inspired algorithm with -hill climbing," *Applied Intelligence*, vol. 48, no. 11, pp. 4429–4447, 2018.
- [10] N. Saini, S. Saha, A. Harsh, and P. Bhattacharyya, "Sophisticated som based genetic operators in multi-objective clustering framework," *Applied Intelligence*, pp. 1–20, 2018.
- [11] M. Mavrovouniotis, C. Li, and S. Yang, "A survey of swarm intelligence for dynamic optimization: Algorithms and applications," *Swarm and Evolutionary Computation*, vol. 33, pp. 1–17, 2017.
- [12] T. Cura, "A particle swarm optimization approach to clustering," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1582–1588, 2012.
- [13] K. K. Bharti and P. K. Singh, "Chaotic gradient artificial bee colony for text clustering," *Soft Computing*, vol. 20, no. 3, pp. 1113–1126, 2016.



- [14] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, no. 2, pp. 495–513, 2016.
- [15] A. Fathy and H. Rezk, "Multi-verse optimizer for identifying the optimal parameters of pemfc model," *Energy*, vol. 143, pp. 634–644, 2018.
- [16] P. Kumar, S. Garg, A. Singh, S. Batra, N. Kumar, and I. You, "Mvo-based two-dimensional path planning scheme for providing quality of service in uav environment," *IEEE Internet of Things Journal*, 2018.
- [17] A. K. Abasi, A. T. Khader, M. A. Al-Betar, S. Naim, and S. N. a. Alyasseri, Zaid Abdi Alkareem Makhadmeh, "A novel hybrid multi-verse optimizer with k-means for text documents clustering," *Neural Computing and Applications*, 2020.
- [18] H. Faris, M. A. Hassonah, A.-Z. Ala'M, S. Mirjalili, and I. Aljarah, "A multi-verse optimizer approach for feature selection and optimizing svm parameters based on a robust system architecture," *Neural Computing and Applications*, pp. 1–15, 2017.
- [19] I. Benmessahel, K. Xie, and M. Chellal, "A new evolutionary neural networks based on intrusion detection systems using multiverse optimization," *Applied Intelligence*, pp. 1–13, 2017.
- [20] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for k-medoids clustering," *Expert systems with applications*, vol. 36, no. 2, pp. 3336–3341, 2009.
- [21] A. Huang, "Similarity measures for text document clustering," *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008)*, pp. 49–56, 2008.
- [22] A. I. Kadhim, Y.-N. Cheah, and N. H. Ahamed, "Text document preprocessing and dimension reduction techniques for text document clustering," *4th International Conference on Artificial Intelligence with Applications in Engineering and Technology (ICAIET)*, pp. 69–73, 2014.
- [23] R. Zhao and K. Mao, "Fuzzy bag-of-words model for document representation," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 794–804, 2018.
- [24] K. K. Bharti and P. K. Singh, "Opposition chaotic fitness mutation based adaptive inertia weight bpsa for feature selection in text clustering," *Applied Soft Computing*, vol. 43, pp. 20–34, 2016.
- [25] J. Singh and V. Gupta, "A systematic review of text stemming techniques," *Artificial Intelligence Review*, vol. 48, no. 2, pp. 157–217, 2017.
- [26] M. N. P. Katariya, M. Chaudhari, B. Subhani, G. Laxminarayana, K. Matey, M. A. Nikose, S. A. Tinkhede, and S. Deshpande, "Text preprocessing for text mining using side information," *International Journal of Computer Science and Mobile Applications*, vol. 3, no. 1, pp. 01–05, 2015.
- [27] D. Janiga, R. Czarnota, J. Stopa, P. Wojnarowski, and P. Kosowski, "Performance of nature inspired optimization algorithms for polymer enhanced oil recovery process," *Journal of Petroleum Science and Engineering*, vol. 154, pp. 354–366, 2017.
- [28] W. Song, Y. Qiao, S. C. Park, and X. Qian, "A hybrid evolutionary computation approach with its application for optimizing text document clustering," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2517–2524, 2015.
- [29] D. Mustafi and G. Sahoo, "A hybrid approach using genetic algorithm and the differential evolution heuristic for enhanced initialization of the k-means algorithm with applications in text clustering," *Soft Computing*, pp. 1–18, 2018.
- [30] Z. A. A. Alyasseri, A. T. Khader, M. A. Al-Betar, A. K. Abasi, and S. N. Makhadmeh, "EEG signals denoising using optimal wavelet transform hybridized with efficient metaheuristic methods," *IEEE Access*, vol. 8, pp. 10 584–10 605, 2019.
- [31] M. A. Al-Betar, O. A. Alomari, and S. M. Abu-Romman, "A triz-inspired bat algorithm for gene selection in cancer classification," *Genomics*, vol. 112, no. 1, pp. 114–126, 2020.
- [32] S. N. Makhadmeh, A. T. Khader, M. A. Al-Betar, S. Naim, A. K. Abasi, and Z. A. A. Alyasseri, "Optimization methods for power scheduling problems in smart home: Survey," *Renewable and Sustainable, Energy Reviews*, vol. 115, 2019.