Kufa Journal of Engineering
Vol. 13, No. 2, April 2022, P.P. 28-42
Article history: Received date 6 September 2021, accepted date 6 February 2022

AN IMPROVED MULTI-VERSE OPTIMIZER FOR TEXT DOCUMENTS CLUSTERING

Ammar K. Abasi[1,2], Ahamad T. Khader [2] and Mohammed A. Al-Betar[1,3]

[1] Artificial Intelligence Research Center (AIRC), College of Engineering and Information Technology, Ajman University, United Arab Emaraties. Email: ammar_abasi@student.usm.my

[2] School of Computer Sciences, Universiti Sains Malaysia, 11800, Penang, Malaysia. Email: tajudin@usm.my

[3] Department of information technology, Al-Huson University College, Al-Huson, Irbid-Jordan. Email: m.albetar@ajman.ac.ae

HTTPS://DOI.ORG/10.30572/2018/KJE/130203

ABSTRACT

Text document clustering (TDC) represents a key task in text mining and unsupervised machine learning, which partitions a specific documents' collection into varied K-groups according to certain similarity/dissimilarity criterion. The multi-verse optimizer algorithm (MVO) is a stochastic population-based algorithm, which was recently introduced and successfully utilized to tackle many optimization problems that are complex. The original MVO performance is limited to the utilization of only the best solution in the exploitation phase (local search capability), which makes it suffer from entrapment in local optima and low convergence rate. This paper aims to propose a novel method of modifying the MVO algorithm called link-based multi-verse optimizer algorithm (LBMVO) to enhance the exploitation phase in the original MVO. Generally, LBMVO has outperformed or at least showed that it is profoundly competitive compared with the original MVO algorithm.

KEYWORDS: Multi-verse optimizer; Optimization; Test clustering; Data clustering; Neighborhood selection strategy

Article information: This manuscript was handled by Abbas Al-Hdabi, Managing Editor.

1. INTRODUCTION

Text documents clustering (TDC) is one of the powerful and efficient unsupervised learning techniques in text mining. it represents, in general, the first step of Topic Extraction (TE) to identify the documents, which address a related subject matter. TDC aims to divide documents into groups (also called clusters), where similar documents are placed in the same cluster and dissimilar documents in different clusters. This technique helps construct meaningful partitions of massive amounts of heterogeneous digital documents. Partition text documents clustering (PTDC) is defined by (Abasi et al., 2021a) as follows, "the process of partitioning a collection of documents into several sub-collections based on their similarity of contents." TDC has been widely studied in the last few years due to two main reasons. First, it is difficult to assign humongous documents manually to extract meaningful information. Second, to avoid personal biases in judging documents that belong to any filed or category (Abasi et al., 2020a). According to (Abasi et al, 2020a), even human experts in automatic clustering do not intervene, which entails that there is no need for any prior knowledge about the texts (i.e., without consulting class label of the documents).

Generally, each document in TDC is represented as a vector using the vector space model (VSM). A widely used approach for document representation is a bag of words (Katrawi et al., 2020), where each distinct term that is present in the documents' collection is considered as a feature for the documents' representation. Therefore, a document is represented by a multi-dimensional feature space, where the cell value of each dimension corresponds to a weighted value, e.g., term frequency inverse document frequency (Tf-Idf) (Katrawi et al., 2021), of the concerned term within the document. Hundreds of thousands of informative and uninformative features (i.e., irrelevant, redundant, and noisy features) originate from the transformation process

Over two decades, substantial efforts were exerted by metaheuristic algorithms to solve TDC because the existing deterministic approaches exhibited powerlessness in finding globally optimal solutions to such problems. Most of the algorithms were originated from the evolutionary-based algorithms (EA) survival of the fittest theory, trajectory-based algorithms (TAs), and swarm intelligence (SI) (Alyasseri et al., 2021a). In general, all the metaheuristics are nature-inspired (i.e., inspired by ethology, biology or physics). Their components exhibit stochastic behaviors (involving random variables) and they set many parameters that need to be adapted to the problem (Alyasseri et al., 2021b; Makhadmeh et al., 2019a; Makhadmeh et al., 2021).

Evolutionary algorithms are the algorithms that mimic evolutionary processes in their nature. These algorithms are grounded on the survival of the fittest candidate for a specific environment. They begin with a specific population (i.e., a set of solutions) that endeavors to survive in a specific environment (and are defined with fitness evaluation). The parent population works on sharing its adaptation properties in an environment with children having various evolution mechanisms like genetic crossover and mutation. This process remains for several generations (i.e., iterative process) until the most suitable solutions are obtained for an environment. Some of these evolutionary algorithms include GA, Evolution Strategies (ES), Genetic Programming (GP), Differential Evolution (DE), and Biogeography-Based Optimization (BBO) (Alomari et al., 2021).

TA is an algorithm, which is initiated by a single solution. Iteration by iteration, this solution undergoes improvements using neighboring-moves operators until the most optimal local solution in a similar search space region of the initial solution can be obtained. Although TAs can deeply search the search space region of the initial solution and reach the local optima, they fail to navigate some search space regions simultaneously. The main TAs, which are utilized for TDC, include K-means and K-medoids, whereas other TAs, which are used for TDC, include β-hill climbing (Abasi et al., 2019) and self-organizing maps (SOM).

SI represents the natural metaheuristics group, which is inspired by 'swarms collective intelligence'. This collective intelligence is built via a homogeneous agents' population; these agents can interact with one another, and they can interact with the environment as well. Good examples of this intelligence can be found in the ants' colonies and flocks of birds, as well as schools of fish, etc. The PSO (Abasi et al., 2020b) is developed following the birds' swarm behavior. The firefly algorithm (FA) (Abasi et al., 2019) is formulated according to the fireflies' flashing behavior, whereas the Bat Algorithm (BA) (Makhadmeh et al., 2020) depends on the bats' echolocation behavior.

The rest of this research is organized into the following sections. The second section reviews previous studies and works. The third section describes in detail the proposed methodology. The fourth section provides the analysis, as well as the empirical results in relation to the efficient performance of the proposed method. The conclusions, as well as recommendations, are provided in the fifth section of this paper.

2. LITERATURE REVIEW

Over two decades, substantial efforts were exerted by metaheuristic algorithms to solve TDC because the existing deterministic approaches exhibited powerlessness in finding globally optimal solutions to such problems. Most of the algorithms were originated from the evolutionary-based algorithms (EA) survival of the fittest theory, trajectory-based algorithms (TAs), and swarm intelligence (SI) (Alyasseri et al., 2019a) In general, all the metaheuristics are nature-inspired (i.e., inspired by ethology, biology or physics). Their components exhibit stochastic behaviors (involving random variables) and they set many parameters that need to be adapted to the problem (Makhadmeh et al., 2019b).

Evolutionary algorithms are the algorithms that mimic evolutionary processes in their nature. These algorithms are grounded on the survival of the fittest candidate for a specific environment. They begin with a specific population (i.e., a set of solutions) that endeavors to survive in a specific environment (and are defined with fitness evaluation). The parent population works on sharing its adaptation properties in an environment with children having various evolution mechanisms like genetic crossover and mutation. This process remains for several generations (i.e., iterative process) until the most suitable solutions are obtained for an environment. Some of these evolutionary algorithms include GA, Evolution Strategies (ES) (Abasi et al., 2021b), Genetic Programming (GP) (Abasi et al., 2020c), Differential Evolution (DE) (Abasi et al., 2021c), and Biogeography-Based Optimization (BBO) (Alyasseri et al., 2019b).

TA is an algorithm, which is initiated by a single solution. Iteration by iteration, this solution undergoes improvements using neighboring-moves operators until the most optimal local solution in a similar search space region of the initial solution can be obtained. Although TAs can deeply search the search space region of the initial solution and reach the local optima, they fail to navigate some search space regions simultaneously. The main TAs, which are utilized for TDC, include K-means and K-medoids, whereas other TAs, which are used for TDC, include β-hill climbing and self-organizing maps (SOM) (Alyasseri et al., 2020).

SI represents the natural metaheuristics group, which is inspired by ‘swarms collective intelligence’. This collective intelligence is built via a homogeneous agents’ population; these agents can interact with one another, and they can interact with the environment as well. Good examples of this intelligence can be found in the ants' colonies and flocks of birds, as well as schools of fish, etc. The PSO (Alyasseri et al., 2020) is developed following the birds' swarm behavior. The firefly algorithm (FA) is formulated according to the fireflies' flashing behavior, whereas the Bat Algorithm (BA) depends on the bats' echolocation behavior.

3. METHODOLOGY

Since 2016, various attempts have been carried out and numerous variants have, accordingly, been proposed with the aim of improving the MVO performance (Abasi et al., 2020d). Also, based on the theorem of no free lunch, there does not exist an equally successful global optimizer, which can provide solutions to all the optimization problems. From such a perspective, it is necessary to keep the evolutionary algorithms diverse to investigate the inverse problems. MVO, like other optimization methods, try to achieve the right balance between exploration and exploitation during the search. The exploration is the ability of the MVO to navigate several search space regions. The randomness is normally used to empower the exploration through wormhole existence probability (WEP) parameter. The exploitation is the ability of MVO to make use of accumulative search through deeply search in each search space region to which it converges. This is done by using the features of the best solution found during the search using travelling distance rate (TDR) parameter. The MVO has several inherent drawbacks such as the convergence that is premature, and it can be trapped within local. This study focused on the exploitation phase because the solutions in the original MVO learn only from the best solution regardless of other useful solutions in the search.

As mentioned earlier in the procedure of MVO, the search cannot only be around the best solution in the exploitation phase. Conventionally, it is recognized in the field of optimization that global optima are possibly reached by not concentrating on the best solution only; however, further solutions can be worse compared with the best solutions via sharing their own good attributes in the searching path for the global optima.

This represents the principal MVO drawback as the heavy focus is on the best solution regardless of other useful solutions in the search. The suboptimal are sometimes close to the global optimum; the trapped individuals' neighborhoods might comprise the global optimum. It is helpful, in this case, to search the individuals' neighborhoods so that better solutions can be found. In any optimization algorithm, the neighborhood selection represents a crucial step because it exerts an effect on the convergence speed, as well as the final solution quality. The suggested NSS can be classified into three phases. These include providing the neighbor information matrix for the solutions' population in the first phase. In the second phase, the Link function is computed by utilizing Eq.3.2. In the third phase, the rank of each of the solutions is calculated via Link matrix; the neighborhood of each of the solutions are selected according to ranks.

3.1. Neighbors and Link Function

The neighbors of a solution Ui in the population matrix are the solutions, which are similar to Abasi et al., (2020d). Let sim (Ui,Uj) represents a similarity function capturing the pairwise similarity between two solutions, Ui,Uj, having values between 0 and 1, with a larger value to indicate higher similarity. For a certain threshold Θ, Ui and Uj can be defined as neighbors next to each other if:

$$\text{sim}\ (Ui,Uj) \geq \Theta, \text{ with } 0 \leq \Theta \leq 1. \qquad 1$$

Here, Θ signifies a user-defined threshold for controlling how alike a solutions' pair should be to be regarded as neighbors next to each other. When the Euclidean distance is used as sim and set Θ to 1, a document is forced to be a neighbor of further identical solutions only. However, when Θ is set to 0, any solutions' pair will be neighbors. The user can select an appropriate value for Θ based on the application.

The information concerning every solution's neighbors in a population is characterized by a neighbor matrix. The neighbor matrix of a population of n solutions represents a n× n symmetric matrix M, in which an entry M (i, j) represents 1 or 0 according to whether solutions Ui, as well as Uj represent neighbors or not. Fig. 1, for example, shows a neighbor matrix of a population U comprising 7 solutions, U1, U2, U3, U4, U5, U6, and U7. The total number of neighbors for the solution Ui in the population represents the summation of 1's in the row i of the matrix M. For example, the number of neighbors for solution U1 equal to 3 (i.e., sum of ones for U1).

M=

	U1	U2	U3	U4	U5	U6	U7	Link
U1	1	0	1	0	0	0	1	3
U2	0	1	0	0	1	0	0	2
U3	1	0	1	0	1	1	0	4
U4	0	1	0	1	0	0	1	3
U5	0	1	1	0	1	0	0	3
U6	0	0	1	0	0	1	0	2
U7	1	0	0	1	0	0	1	3

Fig. 1. An example of compute link function based on neighbor similarity matrix M of population U.

The summation is calculated based on the link function link (Ui,Uj) between two solutions as formulated in Eq. 2.

$$\text{link}\ (Ui,Uj) = \sum_{m=1}^{n} M(i,m) \times M(m,j) \quad (2)$$

After the common neighbors are obtained, the rank values of each pair of the solution are calculated utilizing the link (Ui,Uj). The solutions that have a higher number of common neighbors obtain higher rank values; otherwise, they obtain smaller rank values. The neighbor, which influences the particle the most can be evaluated by ranking. The link function utilizes the neighbor solutions' knowledge in assessing the relationship between a couple of solutions and it is regarded as one efficient method among others to measure how close the two solutions can be. In accordance with this concept, this approach can be used to search neighborhoods in MVO with the aim of improving the local search (exploitation) capability of the MVO in text clustering.

3.2. The Proposed Method for TDC

To maintain the diversity of the solutions in the MVO without significantly slowing down convergence, an NSS in LBMVO has been proposed. As an alternative to learning from the best solution in the exploitation phase, LBMVO learns from the best solution and the best neighbor solution, which is selected by NSS in an early stage with the aim of enhancing diversity. The wormhole channels' mechanism is updated in LBMVO and is added as follows:

$$x_i^j = \begin{cases} \begin{cases} bestNSj + TDR \times ((ubj - lbj) \times r4 + lbj) & r3 < 0.5,\ r2 < WEP \\ bestNSj - TDR \times ((ubj - lbj) \times r4 + lbj) & r3 \geq 0.5 \end{cases} \\ x_i^j & r2 \geq WEP \end{cases} \quad (3)$$

where bestNSj shows jth variable of best neighbor solution obtained, ubj is the upper bound of jth variable, lbj shows the lower bound of jth variable, r3 and r2 are a random numbers between 0 and 1.

Fig. 2 shows an example of selecting the best solution of the TDC problem using LBMVO, the final result of this phase is an optimized solution x = (x1, x2, . . ., xd), which is the optimal organizing unsupervised text documents. The steps of LBMVO are summarized as follows:

1- Initialize a set of feasible solutions based on the nature of the TDC problem parameters (i.e., the number of the document, clusters, etc.) using the generation function.

2- Calculate the objective function f (x) (i.e., Euclidian distance) for all solutions.

3- Sort all solutions based on the objective function value from best to worst and determine the current best solution and the best neighbor solution selected by NSS.

4- Update each solution based on White/Black and Worm Holes for each decision variable i in the current solution (xi):

Generate a random value of r1 and compare it with the objective function value to manipulate the current decision variable with a global search (exploration) by two cases:

> case1: Replace with the value from its historical solutions (better solutions). (see example: (current solution 2, decision variable 2), (current solution 3, decision variable 1), and (current solution 3, decision variable 4)).
>
> case2: the value remains unchanged (see example: (current solution 2, decision variable 3), (current solution 2, decision variable 7), and (current solution 3, decision variable 10)).

Generate a random value r2 and compare it with WEP to manipulate the current decision variable with local search (exploitation) by two cases:

> Case1: Replace with the value from the best solution (see the example: (current solution 2, decision variable 1)).
>
> Case2: Replace with the value from the best neighbor solution (see the example: (current solution 2, decision variable 2), and (current solution 3, decision variable 8)).
>
> Repeat step 4 for all the solutions and repeat steps 2-4 based on the end criterion that is satisfied.

5- Produce the best solution.

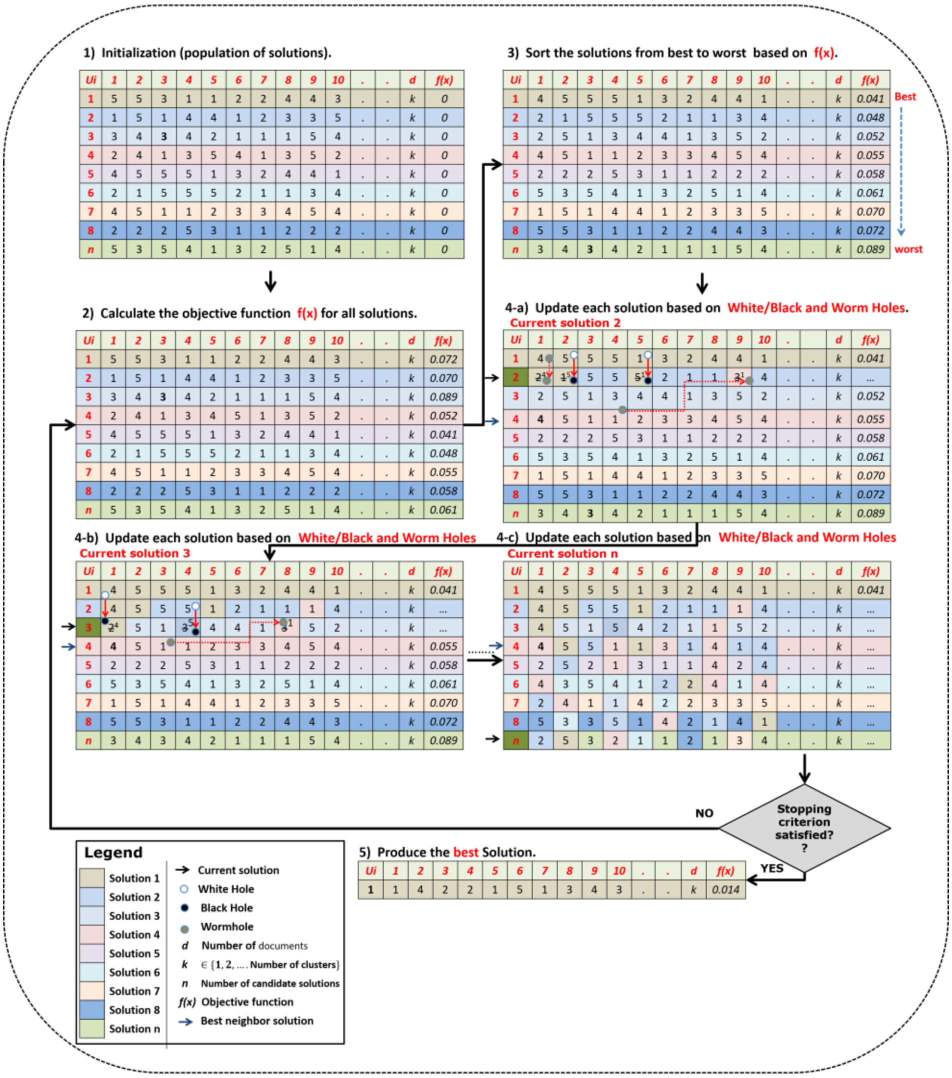

Fig. 2. Steps of proposed LBMVO to improve the solutions over the course of iterations.

In this study, Neighbour Operator (Noperator) is responsible for managing the exploitation phase with a probability range between [0%,100%] of each decision variable i in the solution xi in order to change the decision variable i with decision variable j from best solution or best neighbour solution. Fig. 3 shows the different Noperator values (i.e., 0%,100%,50%,20%) were conducted where the optimal LBMVO is determined. It can be noted that the value of Noperator controls the size of moves to the best neighbour solution of the current one. The higher Noperator value, the higher probability of the current solution changes the decision variable from best neighbour solution. In contrast, the lower Noperator value, the higher probability of the current solution changes the decision variable from best solution (i.e., original MVO).

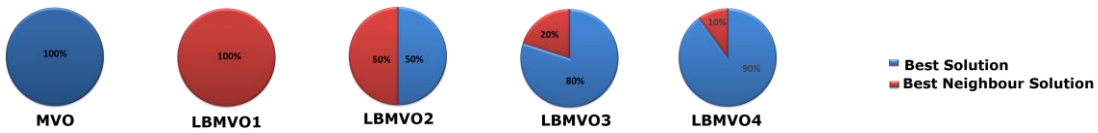

Fig. 3. Selection probability method of the original MVO and LBMVO in the exploitation phase.

4. RESULTS AND DISCUSSION

The experimental design of the proposed LBMVO algorithm in TDC is addressed in this section. A variety of datasets are used to examine the proposed algorithm's performance.

Six external evaluation measures are implemented as it is conventionally done. These measures include accuracy, recall, precision, F-measure, purity, entropy criteria, and the internal quality measure is the sum of distances (the intra-cluster). To achieve a comparative evaluation, the results that were obtained by using the evaluation measures were compared with the results that were obtained by using the original MVO and utilizing the same objective function.

In order to ensure the fairness in the comparison, all methods are re-used and reimplemented using the same parameters and dataset on the six external evaluation measures (i.e., accuracy, recall, precision, F-measure, purity, and entropy). Furthermore, the initial solution/s are the same and the number of evolutions are the same.

This section studies the effect of the link-based selection on the convergence behaviour of MVO. Four version of MVO have been proposed (please refer to Fig. 2): LBMVO1 which is MVO with Link-based Selection that set Noperator = 100%. LBMVO2 which is MVO with Link-based Selection that set Noperator = 50%. LBMVO3 which is MVO with Link-based Selection that set Noperator = 20%. LBMVO4 which is MVO with Link-based Selection that set Noperator = 10%. Recall, Noperator is the percentage of using Link-based Selection in the MVO. The results are compared against original MVO. The results are obtained in terms of six measures include accuracy, re-call, precision, F-measure, purity, and entropy. The obtained results of each measure are the average of 30 replicated runs.

The experimental results showed that the proposed LBMVO3 algorithms work effectively in solving the TDC problem. The evaluation was carried out according to the algorithm's performance, and it was compared with other comparative algorithms. The results over six TCD are summarized based on accuracy, recall, precision, F-measure, purity, entropy as shown in Table1. The best results are highlighted in bold font. The proposed LBMVO3 achieved the best-recorded results in nine datasets from a total of sixteen datasets based on the accuracy measure (i.e., DS1, DS2, DS4, and DS6), followed by LBMVO2. Based on the evaluation measure, which represents one commonly used standard measure, among others, in the text clustering domain, the embed NSS in the searching strategy of MVO was effective compared to the original MVO algorithms, as well as the comparative algorithms.

According to the F-measure, LBMVO3 achieved the best results in eight datasets from a total of eleven datasets (i.e., DS1, DS2, DS4, and DS6), followed by LBMVO2. According to the measures, the proposed method algorithm has exhibited successful outcomes, as well as efficient comparative performance. The LBMVO3 achieved almost all the best results based on the F-measure in comparison with other comparative algorithms. Such a result is based on the purity measure to support the proposed method. LBMVO3 represents an effective algorithm in solving the text clustering problem compared to the original MVO algorithms, as well as other comparative algorithms. The implementation of the neighborhood strategy in the exploitation phase of the MVO algorithm, therefore, enhanced the local search capability. A proper balance between exploration, as well as exploitation, enhanced the proposed MVO algorithm's performance so that accurate clusters are obtained.

According to the Accuracy, the modified version of MVO (i.e., LBMVO3) achieved very competitive results. Also, it outperformed other algorithms for both clustering techniques and optimization algorithms, which proved that it can embed NSS in the MVO searching strategy to enhance the exploitation capability performance. This is because LBMVO3 in the exploitation phase depends on 80% of learns from best solution and 20% from best neighbor solution. The search was, therefore, more efficient towards the global optimum.

Table 1. Results of accuracy, precision, recall, F-measure, purity, and entropy for six text clustering datasets.

Dataset	Measure	Algorithms				
		MVO	LBMVO	LBMVO2	LBMVO	LBMVO
DS1	Accuracy	0.459	0.4635	0.4653	**0.4747**	0.4326
	Precision	0.571	0.5770	0.5836	**0.5876**	0.5711
	Recall	0.482	0.4867	0.4866	**0.4971**	0.4745
	F-measure	0.524	0.5315	0.5366	**0.5435**	0.4608
	Purity	0.568	0.5707	0.5776	**0.5823**	0.4993
	Entropy	0.520	0.5020	0.4829	**0.4763**	0.6104
	Rank	4	3	2	1	5
DS2	Accuracy	0.404	0.3803	0.4092	0.4405	**0.4915**
	Precision	0.439	0.4154	0.4451	**0.4740**	0.4166
	Recall	0.384	0.3590	0.3870	**0.4146**	0.3873
	F-measure	0.410	0.3926	0.4124	**0.4378**	0.3907
	Purity	0.434	0.4110	0.4385	**0.4707**	0.4525
	Entropy	0.712	0.7175	0.6921	**0.6843**	0.7509
	Rank	4	5	2	1	3
DS3	Accuracy	0.448	0.4113	0.4626	**0.4901**	0.4086
	Precision	0.507	0.4708	0.5175	**0.5413**	0.4219
	Recall	0.439	0.4005	0.4493	**0.4801**	0.3986
	F-measure	0.470	0.4281	0.4800	**0.5100**	0.4384
	Purity	0.544	0.5042	0.5546	**0.5833**	0.5426
	Entropy	0.522	0.5362	0.5077	**0.4978**	0.5772
	Rank	3	4	2	1	5

Dataset	Measure	Algorithms				
		MVO	LBMVO1	LBMVO2	LBMVO3	LBMVO4
DS4	Accuracy	0.4630	0.4507	0.4724	**0.5034**	0.4934
	Precision	0.4569	0.4430	0.4662	**0.4981**	0.4808
	Recall	0.4419	0.4220	0.4591	0.4839	**0.5585**
	F-measure	0.4569	0.4371	0.4742	**0.5007**	0.4562
	Purity	0.6081	0.5916	0.6205	**0.6493**	0.4503
	Entropy	0.5355	0.6236	0.5186	**0.4951**	0.5179
	Rank	4	5	2	1	3
DS5	Accuracy	0.5291	0.5177	0.5235	0.5592	**0.6474**
	Precision	0.5213	0.5133	0.5214	0.5550	**0.5651**
	Recall	0.4496	0.4438	0.4483	**0.4777**	0.4584
	F-measure	0.4831	0.4742	0.4831	**0.5111**	0.4866
	Purity	0.6069	0.6037	0.6012	**0.6359**	0.5428
	Entropy	0.6625	0.6591	0.6498	0.6260	**0.5784**
	Rank	4	5	3	2	1
DS6	Accuracy	0.7043	0.6776	0.7316	**0.7499**	0.6258
	Precision	0.6919	0.6628	0.7240	**0.7470**	0.6138
	Recall	0.6844	0.6546	0.7188	**0.7318**	0.6485
	F-measure	0.6881	0.6553	0.7204	**0.7420**	0.6218
	Purity	0.6742	0.6489	0.7012	**0.7237**	0.5767
	Entropy	0.5113	0.5314	0.4622	**0.4400**	0.4627
	Rank	3	4	2	1	5

Furthermore, comparing with (Abasi et al., 2020d) the results showed that clustering algorithms that are based on the metaheuristic algorithms, i.e., basic harmony search (HS), genetic algorithm (GA), particle swarm optimization (PSO), krill herd algorithm (KHA), covariance matrix adaptation evolution strategy (CMAES), coyote optimization algorithm (COA), as well as original MVO performed comparatively in a better manner compared with the conventional clustering techniques in most cases. This might be attributed to the clustering techniques' inefficiency in exploring the specific search space. A solution is generated near the centroids that are initially selected; even its multiple independent runs cannot solve this problem. The population-based algorithms' characteristic involves a simultaneous exploration of varied search space regions in synergy. They can, therefore, escape from a local optimum solution to obtain quite a better solution via an efficient search of the search space. Future studies can investigate the applications of the proposed approach in solving some other real-life problems like scientific articles' classification and web documents, topic extraction, automatic grading of essays, etc., in addition to the hybridized MVO with local search strategies so that the initial solutions and the exploitation capability during the optimization process are enhanced.

5. CONCLUSION

A novel multi-verse optimizer algorithm called LBMVO is proposed in this paper with the integration of a new neighbor selection strategy to solve the text document clustering problem. The proposed algorithm introduces a new updating strategy to learn from the neighbor best solution, as well as the best solution to improve the local search capability of the original MVO in the exploitation phase. The proposed algorithm generates many diverse solutions with searching more cavernous around the candidate solutions, which may contain the global optimum. The LBMVO proved efficient as it can automatically partition sixteen benchmark datasets. The quality of the obtained results was assessed using six measures: accuracy, precision, recall, F-measure, purity, and entropy. The results were compared with the original MVO. Based on the results, it was found that the proposed approach can reach the global optimal solution for the entire datasets, while other algorithms were trapped at local optima. The results revealed that the proposed algorithm is suitable for partitioning the datasets in an automated manner by making the clusters more heterogeneous. They found that the LBMVO has significantly increase the performance of six external measures as well as high convergence rate, for example, The LBMVO3 based on F-measure achieved the best results in five datasets (i.e., DS1, DS2, DS3, DS4, and DS6) from a total of six datasets.

6. REFERENCES

Abasi, A. K., Khader, A. T., Al-Betar, M. A., Alyasseri, Z. A. A., Makhadmeh, S. N., Al-laham, M., & Naim, S. (2021a). A Hybrid Salp Swarm Algorithm with β-Hill Climbing Algorithm for Text Documents Clustering. *Evolutionary Data Clustering: Algorithms and Applications*, 129.

Abasi, A. K., Khader, A. T., Al-Betar, M. A., Naim, S., Makhadmeh, S. N., & Alyasseri, Z. A. A. (2021b). A novel ensemble statistical topic extraction method for scientific publications based on optimization clustering. *Multimedia Tools and Applications*, *80*(1), 37–82.

Abasi, A. K., Khader, A. T., Al-Betar, M. A., Naim, S., Alyasseri, Z. A. A., & Makhadmeh, S. N. (2021c). An ensemble topic extraction approach based on optimization clusters using hybrid multi-verse optimizer for scientific publications. *Journal of Ambient Intelligence and Humanized Computing*, *12*(2), 2765–2801.

Abasi, A. K., Khader, A. T., Al-Betar, M. A., Naim, S., Alyasseri, Z. A. A., & Makhadmeh, S. N. (2020a). A novel hybrid multi-verse optimizer with K-means for text documents clustering. *Neural Comput. Appl.*, *32*(23), 17703–17729.

Abasi, A. K., Khader, A. T., Al-Betar, M. A., Naim, S., Makhadmeh, S. N., & Alyasseri, Z. A. A. (2020b). Link-based multi-verse optimizer for text documents clustering. *Applied Soft Computing Journal*, *87*. https://doi.org/10.1016/j.asoc.2019.106002

Abasi, A. K., Khader, A. T., Al-Betar, M. A., Naim, S., Makhadmeh, S. N., & Alyasseri, Z. A. A. (2020c). An improved text feature selection for clustering using binary grey wolf optimizer. *Proceedings of the 11th National Technical Seminar on Unmanned System Technology 2019*, 503–516.

Abasi, A. K., Khader, A. T., Al-Betar, M. A., Naim, S., Awadallah, M. A., & Alomari, O. A. (2020d). Text documents clustering using modified multi-verse optimizer. *International Journal of Electrical and Computer Engineering*, *10*(6), 6361–6369.

Abasi, A. K., Khader, A. T., Al-Betar, M. A., Naim, S., Makhadmeh, S. N., & Alyasseri, Z. A. A. (2019). A text feature selection technique based on binary multi-verse optimizer for text clustering. *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, 1–6.

Alyasseri, Z. A. A., Abasi, A. K., Al-Betar, M. A., Makhadmeh, S. N., Papa, J. P., Abdullah, S., & Khader, A. T. (2021a). EEG-Based Person Identification Using Multi-Verse Optimizer as Unsupervised Clustering Techniques. *Evolutionary Data Clustering: Algorithms and Applications*, 89.

Alyasseri, Z. A. A., Al-Betar, M. A., Doush, I. A., Awadallah, M. A., Abasi, A. K., Makhadmeh, S. N., Alomari, O. A., Abdulkareem, K. H., Adam, A., Damasevicius, R., & others. (2021b). Review on COVID-19 diagnosis models based on machine learning and deep learning approaches. *Expert Systems*, e12759.

Alyasseri, Z. A. A., Khader, A. T., Al-Betar, M. A., Abasi, A. K., & Makhadmeh, S. N. (2020). EEG Signal Denoising Using Hybridizing Method Between Wavelet Transform with Genetic Algorithm. *Proceedings of the 11th National Technical Seminar on Unmanned System Technology 2019*, 449–469.

Alyasseri, Z. A. A., Khadeer, A. T., Al-Betar, M. A., Abasi, A., Makhadmeh, S., & Ali, N. S. (2019a). The effects of EEG feature extraction using multi-wavelet decomposition for mental tasks classification. *Proceedings of the International Conference on Information and Communication Technology*, 139–146.

Alyasseri, Z. A. A., Khader, A. T., Al-Betar, M. A., Abasi, A. K., & Makhadmeh, S. N. (2019b). EEG signals denoising using optimal wavelet transform hybridized with efficient metaheuristic methods. *IEEE Access*, *8*, 10584–10605.

Alomari, O. A., Makhadmeh, S. N., Al-Betar, M. A., Alyasseri, Z. A. A., Doush, I. A., Abasi, A. K., Awadallah, M. A., & Zitar, R. A. (2021). Gene selection for microarray data classification based on Gray Wolf Optimizer enhanced with TRIZ-inspired operators. *Knowledge-Based Systems*, *223*, 107034.

Katrawi, A. H., Abdullah, R., Anbar, M., & Abasi, A. K. (2020). Earlier stage for straggler detection and handling using combined CPU test and LATE methodology. *International Journal of Electrical and Computer Engineering*, *10*(5), 4910.

Katrawi, A. H., Abdullah, R., Anbar, M., AlShourbaji, I., & Abasi, A. K. (2021). Straggler handling approaches in mapreduce framework: a comparative study. *International Journal of Electrical & Computer Engineering (2088-8708)*, *11*(1).

Makhadmeh, S. N., Khader, A. T., Al-Betar, M. A., Naim, S., Abasi, A. K., & Alyasseri, Z. A. A. (2019a). Optimization methods for power scheduling problems in smart home: Survey. *Renewable and Sustainable Energy Reviews*, *115*, 109362.

Makhadmeh, S. N., Khader, A. T., Al-Betar, M. A., Naim, S., Alyasseri, Z. A. A., & Abasi, A. K. (2019b). Particle swarm optimization algorithm for power scheduling problem using smart battery. *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, 672–677.

Makhadmeh, S. N., Khader, A. T., Al-Betar, M. A., Naim, S., Abasi, A. K., & Alyasseri, Z. A. A. (2021). A novel hybrid grey wolf optimizer with min-conflict algorithm for power scheduling problem in a smart home. *Swarm and Evolutionary Computation*, *60*, 100793.

Makhadmeh, S. N., Khader, A. T., Al-Betar, M. A., Naim, S., Alyasseri, Z. A. A., & Abasi, A. K. (2020). A min-conflict algorithm for power scheduling problem in a smart home using battery. *Proceedings of the 11th National Technical Seminar on Unmanned System Technology 2019*, 489–501.