Engineering Applications of Artificial Intelligence 72 (2018) 54–66

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Evolutionary static and dynamic clustering algorithms based on multi-verse optimizer

Sarah Shukri[a], Hossam Faris[a], Ibrahim Aljarah[a,*], Seyedali Mirjalili[b], Ajith Abraham[c]

[a] *King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan*
[b] *Institute for Integrated and Intelligent Systems, Griffith University, Nathan, Brisbane, QLD 4111, Australia*
[c] *Machine Intelligence Research Labs (MIR Labs), Auburn, WA, United States*

ARTICLE INFO

Keywords:
Optimization
Metaheuristics
Multi-verse optimizer
Clustering
Swarm Intelligence
Data mining
Machine learning
Evolutionary Computation

ABSTRACT

Clustering based on nature-inspired algorithms is considered as one of the fast growing areas that aims to benefit from such algorithms to formulate a clustering problem as an optimization problem. In this work, the search capabilities of a recent nature-inspired algorithm called Multi-verse Optimizer (MVO) is utilized to optimize clustering problems in two different approaches. The first one is a static clustering approach that works on a predefined number of clusters. The main objective of this approach is to maximize the distances between different clusters and to minimize the distances between the members in each cluster. In an attempt to overcome one of the major drawbacks of the traditional clustering algorithms, the second proposed approach is a dynamic clustering algorithm, in which the number of clusters is automatically detected without any prior information. The proposed approaches are tested using 12 real and artificial datasets and compared with several traditional and nature-inspired based clustering algorithms. The results show that static and dynamic MVO algorithms outperform the other clustering techniques on the majority of datasets.

1. Introduction

Clustering is a data mining task that aims to classify data instances which are usually represented as vectors of data points into a set of groups (clusters) based on predefined similarity metrics. Traditionally, similarity metrics are represented by distance measurements between the different data points in different clusters (Donalek, 2011). Based on the similarity measures, data instances within the same cluster are expected to be more similar than any instances from other clusters. In contrast, the instances from different clusters are expected to be more dissimilar (Jain et al., 1999). Clustering can be used broadly in many applications such as information retrieval (Jardine and van Rijsbergen, 1971), image processing (Reddy et al., 2007), document categorization (Fung et al., 2003), pattern recognition (Baraldi and Blonda, 1999), and web log clustering (Xie and Phoha, 2001).

Clustering is considered as one of the most challenging problems in data mining. Clustering applies unsupervised learning methods by analyzing and extracting useful information and patterns of the target data without having any prior knowledge about the data labels. Therefore, the general challenging problems that face clustering tasks can be summarized in two points: first, clustering deals with unlabeled data where the actual class is unknown (Donalek, 2011). Second, determining the right number of the clusters without any prior information might be difficult and challenging. Usually, experts in the domain are consulted to help in determining the right number of the clusters.

In general, clustering techniques are classified into two main categories: hierarchical and partitional. In the hierarchical clustering, the clustering process is divided into two main classes: agglomerative hierarchical clustering and divisive hierarchical clustering. In agglomerative clustering, the data points are merged together from different clusters into a single cluster. On the other hand, divisive clustering based methods split a large cluster into smaller clusters using similarity based measures (Donalek, 2011; White and Frenk, 1991; Karypis et al., 1999; Bandyopadhyay and Coyle, 2003).

In contrast to the hierarchical clustering, partitional clustering methods (e.g. k-Means algorithm) divide data points into predefined groups/clusters. The cluster representative centroid is then calculated for each group. After that, all clusters are evaluated using some similarity based measures such as the sum of squared errors (SSE), which is calculated based on the distances between the data points and cluster centroids. The main objective of the partitional clustering methods is

[*] Corresponding author.
E-mail addresses: hossam.faris@ju.edu.jo (H. Faris), i.aljarah@ju.edu.jo (I. Aljarah), seyedali.mirjalili@griffithuni.edu.au (S. Mirjalili), ajith.abraham@ieee.org (A. Abraham).

https://doi.org/10.1016/j.engappai.2018.03.013
Received 18 May 2017; Received in revised form 15 March 2018; Accepted 17 March 2018
Available online 3 April 2018

to minimize the SSE measure (Lee and Antonsson, 2000; Jarboui et al., 2007; Xu and Wunsch, 2005). Partitional clustering methods have been widely applied to diverse domains including image segmentation (Marroquin and Girosi, 1993), intrusion detection (Jianliang et al., 2009), and information retrieval (Bellot and El-Bèze, 1999). Partitional clustering methods have some advantages such as low computation complexity and simplicity. However, they suffer from major drawbacks such as the local optimum problem and the sensitivity to the initial centroids. Therefore, many new algorithms have been developed in literature to overcome such drawbacks. Some of these new methods benefit from the capabilities of nature-inspired metaheuristics.

Nature-inspired metaheuristics are search algorithms mainly inspired by some phenomena in nature such as the biological systems, swarming behavior, physical or chemical systems. These algorithms have gained an increased popularity in the recent years due their superior efficiency and reasonable run time as compared to exact methods when solving large-scale and challenging real-world problems (Yang, 2011). One of the well-regarded recent nature-inspired algorithms is the Multi-Verse optimizer algorithm (MVO). MVO was first implemented and proposed by Mirjalili et al. (2016). MVO is inspired by the abstract concepts of the popular multi-verse theory. Since its proposal, MVO has been effectively used and applied in several applications such as engineering applications (Trivedi et al., 2016; Jangir et al., 2017) and machine learning problems (Faris et al., 2017, 2016; Aljarah et al., 2018; Mafarja et al., 2017).

In this paper, the MVO algorithm is employed to tackle the challenges of clustering problems. Two approaches based on MVO are proposed and developed. The first is a static approach, in which the number of clusters has to be predefined before starting the clustering process. The second version is a dynamic clustering approach, which automatically detects the number of clusters without any prior knowledge about the nature of the dataset. The proposed approaches benefit from the search capabilities and powerful operators of the MVO algorithm to alleviate the drawbacks of the partitional clustering algorithms. To verify the effectiveness of the proposed approaches, they will be evaluated based on real and artificial datasets with different levels of difficulty. They will also be compared to a set of popular clustering approaches in the literature.

The remainder of this paper is organized as follows: Section 2 reviews a wide range of related works in the area of nature-inspired clustering based algorithms. Section 3 describes the preliminaries of the clustering problem and the MVO algorithm. In Section 4 and Section 5, the details of the proposed static and dynamic approaches of the MVO based clustering algorithms are given, respectively. The evaluation measures that are used to evaluate the proposed clustering approaches are listed in Section 6. The experiments and results are presented in Section 7. Finally, Section 8 summarizes the conclusions and future directions of this work.

2. Related works

Data clustering is an important and critical problem in data mining. As such, many algorithms were proposed in the literature to solve clustering problems. The K-Means algorithm is considered as one of the most popular partitioning clustering algorithms (MacQueen et al., 1967). This algorithm aims to partition a given dataset into a pre-defined number of groups (k) by minimizing the distances between the data points in the same group and maximizing the distance between the data points in different groups. Another traditional clustering algorithm is called Furthest First (FF) (Kumar et al., 2013). FF is an enhanced version of K-Means algorithm that assigns each new centroid at furthermost location from the existing centroids. This technique speeds up the clustering process compared to the original K-Means algorithm.

Most of the traditional clustering techniques suffer from several drawbacks including the need to determine the number of the groups as an input parameter, which is not practical in real-world problems, the possibility to fall in a local optimum, and the sensitivity to the initial centroids. To tackle these issues, many nature-inspired algorithms were introduced. For instance, Maulik and Bandyopadhyay (2000) proposed a genetic algorithm-based clustering (CGA) in 2000. The optimization capability of genetic algorithm was used to find the best centroids. The experimental results proved the superiority of the CGA algorithm over the K-Means algorithm. Another variant of the genetic algorithm called grouping genetic algorithm (GGA) in Agustıet al. (2012) was used to solve the clustering problem. GGA applied the concepts of grouping encoding and evolutionary operators (mutation, and crossover) on the clustering. The performance of the algorithm in different clustering problems obtained competitive results. Other works that implemented GA based clustering can be found in Scheunders (1997), Doval et al. (1999), Rao et al. (2016) and Ding and Fu (2016).

In Van der Merwe and Engelbrecht (2003), Particle Swarm Optimization (PSO) was applied on clustering analysis. In this work, the capability of the PSO and its advantages such as avoiding the premature convergence to local optima is used to refine the centroids. In 2008, Ahmadyfard and Modares (2008) combined the PSO and K-Means algorithm. The motivation of this hybrid was to benefit from the fast convergence of the PSO in the initial generations, and K-Means, which is faster to reach global solution. Parallel versions of PSO clustering were introduced in Aljarah and Ludwig (2012, 2013a, c). Hassanzadeh and Meybodi (2012) presented a new clustering approach based on the Firefly algorithm. The Firefly algorithm was combined with K-Means algorithm to optimize the best values of the best centroids of the predefined clusters.

In Lee and Antonsson (2000), evolutionary-based clustering approach was proposed. The authors proposed a new evolutionary algorithm to optimize the number of the best centroids along with the best set of groups. The proposed approach designed to deal with 2D spatial data due to the limitation of graphical representation for multidimensional data. Each cluster's centroid was represented as a pair of (x, y), and each gene represented n number of centers ordered in an ascending way. Their evaluation measures showed promising results.

In Shelokar et al. (2004), a clustering based Ant colony optimization algorithm was proposed to solve clustering problems by simulating the movement technique performed by ants to group similar instances in the same cluster. The performance of the proposed algorithm showed better results compared to other stochastic algorithms.

In Aljarah and Ludwig (2013b), a clustering algorithm (CGSO) based on the nature-inspired algorithm called Glowworm Swarm Optimization (GSO) was introduced. The CGSO was adjusted to fit the data clustering problem and to locate multiple optimal centroids within the search space. The authors used three fitness functions to evaluate the performance of the algorithm. Their conducted experiments showed that the algorithm outperforms four of the most popular algorithms in most of the used datasets.

In Ozturk et al. (2015), the authors introduced a new nature-inspired clustering based on binary artificial bee colony. The main objective was to introduce a new solution generation mechanism for dynamic clustering by considering new similarity measures in an efficient way. More examples of nature-inspired based clustering approaches can be found in Omran et al. (2005), Senthilnath et al. (2011), Hatamlou (2013), Aljarah and Ludwig (2012) and Al-Madi et al. (2014).

In most of the existing nature-inspired based clustering algorithms, the number of predefined clusters is needed in advance to solve clustering problems. However, in several practical applications, the number of the clusters is unavailable before exploring the data set. Another issue is that some of the nature-inspired clustering algorithms suffer from slow convergence and poor clustering quality. In this paper, two approaches are proposed to take advantage of MVO optimization efficiency to tackle the clustering problem to enhance the speed of the convergence and solve the problem of determining the number of clusters in advance.

3. Preliminaries

3.1. Clustering

In machine learning, clustering can be formulated as follows. Suppose a dataset ξ consists of n points; p_1, p_2, …, p_n. Each point is represented with d-dimensions such as $p_i = (p_{i1}, p_{i2}, \ldots, p_{id})$. Given ξ dataset, a clustering algorithm tries to find a set of k clusters such as $C = \{C_1, C_2, \ldots, C_k\}$. Clusters C are represented by centroid points c_j, $j = 1, 2, \ldots, k$. The clustering algorithm tries to minimize some similarity measures such as Sum of Squared Errors (SSE), which is calculated based on the euclidean distances between the data points and the nearest cluster centroids (Lee and Antonsson, 2000). The SSE is calculated using the following equation:

$$SSE = \sum_{j=1}^{k} \sum_{i=1}^{|C_j|} D(c_j, p_i)^2 \tag{1}$$

Where D is the Euclidean Distance between the centroid c_j and ith data point, and is given by the following equation:

$$D(c_j, p_i) = \sqrt{\sum_{r=1}^{d} (c_{jr} - p_{ir})^2} \tag{2}$$

where p_{ir} represents rth dimension of the ith data point that belongs to the cluster of c_j centroid.

3.2. Multi-verse optimization algorithm

Multi-Verse optimization algorithm (MVO) is one of the recent nature-inspired algorithms, which was proposed by Mirjalili et al. (2016). The main inspiration of this algorithm is the theory of multi-verse in astrophysics. In this theory, multiple big bangs create multiple universes, which interact with each other through white holes, black holes, and worm holes. It was discussed in the original paper of MVO that matters transfer between two universes though a tunnel with a white hole on one side and a black hole on another. Black holes attract matters and white holes emit matters. A worm hole also creates a tunnel though time and connects different parts of a universe. Note that the following terms are used throughout this sub-section when presenting the MVO algorithm: a universe represents a solution, an object corresponds to a solution, time indicates a generation/iteration, and inflation rate shows the objective value of a universe.

In MVO, each solution is considered as a universe with a possibility to have white holes, black holes, and worm holes. To improve the quality (objective value) of each solution, it is assumed that matter emitters (white holes) are more likely to appear in a solution with a better objective value. By contrast, matter attractors (black holes) tend to be created in a solution with a worse objective value. With this mechanism, values of variables from good solutions are transferred to poor solutions. This increases the possibility of improving poor solutions and consequently improving the average objective value of all solutions.

The main mathematical model of this algorithm depends on Eqs. (3) and (4) which are described as follows.

$$X_i^j = \begin{cases} X_k^j, & r_1 < NI(U_i) \\ X_i^j, & r_1 \geq NI(U_i) \end{cases} \tag{3}$$

where X_i^j represents the jth object of the ith universe, r_1 is a random number in the range (0,1), $NI(U_i)$ represents the normalized inflation rate of the ith universe and X_k^j represents the jth object of the kth universe.

$$X_i^j = \begin{cases} \begin{cases} (X_j + \mathrm{TDR} * (ub - lb) * r_4 + lb), & r_3 < 0.5 \\ (X_j - \mathrm{TDR} * (ub - lb) * r_4 + lb), & r_3 \geq 0.5, \end{cases} & r_2 < \mathrm{WEP} \\ X_i^j, & r_2 \geq \mathrm{WEP} \end{cases} \tag{4}$$

where X_j represents the jth centroid of the best universe obtained so far, *ub* is the upper bound, *lb* is the lower bound, Traveling Distance Rate (*TDR*) is a coefficient, Wormhole Existence Probability (*WEP*) is also a coefficient, r_2, r_3 and r_4 are random numbers in the range (0, 1).

In addition, the MVO algorithm saves the best solution during optimization and utilizes it to impact on the rest of solutions. It was discussed in the original paper of MVO that there is always wormholes established between the best solution and other solutions so that the variables can be easily exchanged. Once more, this increases the chance of improving the solutions and maintains the best solutions obtained so far during optimization. The best solution is returned at the end of optimization as the best approximation of the global optimum for a given optimization problem.

The aforementioned equations and operators require MVO to exchange variables between the solutions. However, they result in similar exploratory and exploitative patterns if we apply them without any adaptive mechanism. The mechanisms equipped in the MVO to appropriately emphasize exploration and exploitation in different stages of optimization are as follows:

$$\mathrm{WEP} = \min + l * \left(\frac{\max - \min}{L}\right) \tag{5}$$

$$\mathrm{TDR} = 1 - \frac{l^{1/p}}{L^{1/p}} \tag{6}$$

where p represents the exploitation factor and equals 6 in this research.

It can be seen that there are two adaptive variables in MVO: WEP and TDR. WEP increases proportional to the number of iterations to enhance exploitation. TDR is increased over the iterations to have more precise exploitation/local search around the best solution obtained so far.

The MVO is an evolutionary algorithm since universes exchange matters. This corresponds to crossover operator, which is a well-known evolutionary operator in the field of optimization. This causes sudden changes in the resulting universes, promotes exploration of the search landscape, and maintains the diversity of universes in each iteration.

Also, each universe receives some variables randomly from the best universe. This corresponds to mutation, which is again a popular evolutionary operator in evolutionary algorithms. The mutation operator causes slight changes in a solution and promote local search (exploitation). Another evolutionary operator is elitism which maintains the best solution obtained so far over the course of optimization. The elitism in MVO is done by storing the best universe formed so far.

Note that the computational complexity of the MVO algorithm is of $O(tnd)$ where t shows the maximum number of generations, n indicates the number of universes (solutions), and d is the number of variables (dimension) of each universe. Since the cost of objective function depends on the problem, it has not been included in the algorithm complexity here. If the complexity of objective function is known, it should be multiplied by t and n only since the objective value is calculated for each universe in each iteration. In this case, complexity can be expressed as $(tn(cof + d))$ where cof stands for Cost Of Objective function.

4. Static clustering based MVO (SCMVO)

The first proposed clustering approach aims to transform the MVO optimizer into a new partitional clustering-based algorithm, which we name SCMVO. The task of SCMVO algorithm is to classify N data points into K predefined number of clusters. In order to solve clustering problems using MVO, two important points should be addressed: the encoding of the individuals and the selection of the fitness function. To address the first point, the universes are designed to represent the candidate clustering solution centroids, where each universe/solution represents K centroids. The main objective of the SCMVO algorithm is to locate the best universe/solution which contains the best set of centroids. Encoding of the universe will be as a *1-dimensional* array, where each universe U_i is formed of K parts, where K is the number

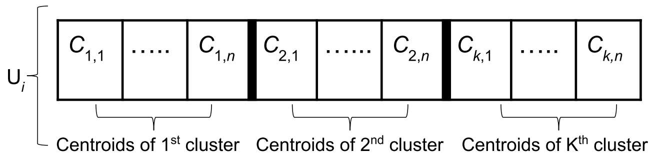


Fig. 1. Universe encoding.

of clusters, and each part of the universe contains n elements, which represent the number of features (variables) in the dataset as shown in Fig. 1.

The main steps of the SCMVO algorithm are discussed as follows:

1. Initialization: The SCMVO algorithm starts with a user-predefined number of clusters (K). Then, SCMVO will randomly initialize the first population of universes by selecting random data points from the dataset itself. The number of random points selected for each universe will be based on the dimension value, and can be calculated as $n \times K$. If a dataset has three clusters and four features, for instance, the dimension of each universe will be 12. If the population consists of 50 universes, then 3 × 50 random data points will be selected from the dataset.
2. Evaluation: In this step, the selected initial centroids will be evaluated. Afterwards, these centroids will be used to partition the data set into K clusters by assigning each data point to the closets cluster. In the next step, distances between data points and clusters' centroids are calculated based on the Euclidean distance. The centroids will be evaluated by calculating the inflation rate for each universe as well, which represents the value of the fitness function. The fitness function that used to evaluate the clustering quality is the SSE, which is calculated by the sum of the distances between each data point and the cluster's centroids by Eq. (1). The objective of the SCMVO algorithm is to minimize the value of the SSE criterion.
3. Update Centroids: In this step, the positions of the centroids are updated using the standard MVO updating equations to ensure the balance between the exploration and exploitation.
4. Termination: SCMVO stops iterating and reports the best found set of centroids, when the end condition is satisfied (*e.g.* exceeding maximum number of iterations). The main processes of SCMVO algorithm is shwon in Fig. 2.

5. Dynamic clustering based MVO (DCMVO)

In this section, a dynamic clustering approach based on MVO (DCMVO) is proposed. Unlike the traditional clustering algorithms, the DCMVO algorithm automatically finds the best number of clusters. DCMVO starts with an arbitrary large number of clusters. Over the course of iterations, the DCMVO algorithm applies a set of processes which includes updating and internal fitness evaluation. It then decreases the number of clusters by one. At the end of each iteration DCMVO uses another fitness evaluation which is called *external fitness function*. The main steps of DCMVO are described as follows:

1. Initialization: The maximum number of clusters is set. Then, DCMVO initializes the first population where each individual is formed by randomly selecting K points from the dataset.
2. Internal fitness: The fitness of each individual is evaluated as in the SCMVO using SSE fitness function which was previously given in Eq. (1).
3. Centroids Update: The centroids are updated using black/white tunnels and worm tunnels as in the SCMVO algorithm.
4. External fitness: At the end of each iteration, DCMVO will assess and evaluate the number of clusters by calculating another evaluation metric. The value of this metric is a combination of the intra-distance and inter-distance. The intra-distance represents the distance between data points within the same cluster, while the inter-distance indicates the distance between centroids across different clusters. This value can be calculated using Eq. (7) (Caliński and Harabasz, 1974).

$$a = (\frac{\text{interdistance}}{K-1}) * (\frac{\text{intradistance}}{n-K}) \tag{7}$$

where K is the maximum number of the clusters that allowed in the solution, and n is the number of data instances. In this paper, K is set to 50 clusters as reported in Caliński and Harabasz (1974).
5. Termination: DCMVO process terminates when the termination condition is met (*e.g.* maximum number of iterations). Note that this number might be equal to the maximum number of clusters. At termination, the algorithm will report K fitness values. The number of clusters that corresponds to the smallest fitness value will be selected along with its centroids, which are found also by DCMVO. The main processes of DCMVO algorithm is shown in Fig. 3.

6. Evaluation measures

The proposed clustering algorithms will be evaluated using two different clustering quality measures: purity and entropy, which are described as follows:

(1) Purity: the percentage of the number of data points that are clustered correctly (Aljarah and Ludwig, 2013b). The purity is calculated using the following equation:

$$\text{Purity} = \frac{1}{n}\sum_{i=1}^{K} |L_i \bigcap C_i| \tag{8}$$

Where n represents the number of data points in the dataset; K represents the number of clusters in the dataset; L_i represents the data points that are truly assigned to the actual cluster and C_i represents the data points that are assigned to cluster i using the clustering algorithm.

(2) Entropy: this clustering quality measure quantifies the semantic distribution of the data points within each cluster (Strehl et al., 2000). The Entropy is calculated using the following equation:

$$\text{Entropy} = \sum_{i=1}^{K} \frac{|C_i|}{n} E(C_i) \tag{9}$$

Where $E(C_i)$ represents the individual entropy of the ith cluster. $E(C_i)$ is given by the following equation:

$$E(C_i) = \frac{-1}{\log(K)} \sum_{i=1}^{k} \frac{|L_i \bigcap C_i|}{C_i} \log(\frac{|L_i \bigcap C_i|}{C_i}) \tag{10}$$

Larger purity values and smaller entropy values indicate better clustering solutions, where the perfect clustering happens when the purity equals 1 and entropy equals 0.

7. Experiments and results

In this section, the SCMVO and DCMVO algorithms are benchmarked using twelve datasets. The SCMVO will be compared with two well-regarded evolutionary algorithms: Clustering based PSO (CPSO) and Clustering based GA (CGA). In addition, the SCMVO will be compared with other traditional clustering algorithms such as K-Means, Hierarchal Clustering (HC), and Farthest First (FF) (Hochbaum and Shmoys, 1985) for verification. The DCMVO algorithm will be compared to the dynamic version of the algorithm based on dynamic PSO and dynamic GA (*i.e.* DPSO and DGA).

The details of the experimental setup, datasets description, and the results are discussed in the following subsections.

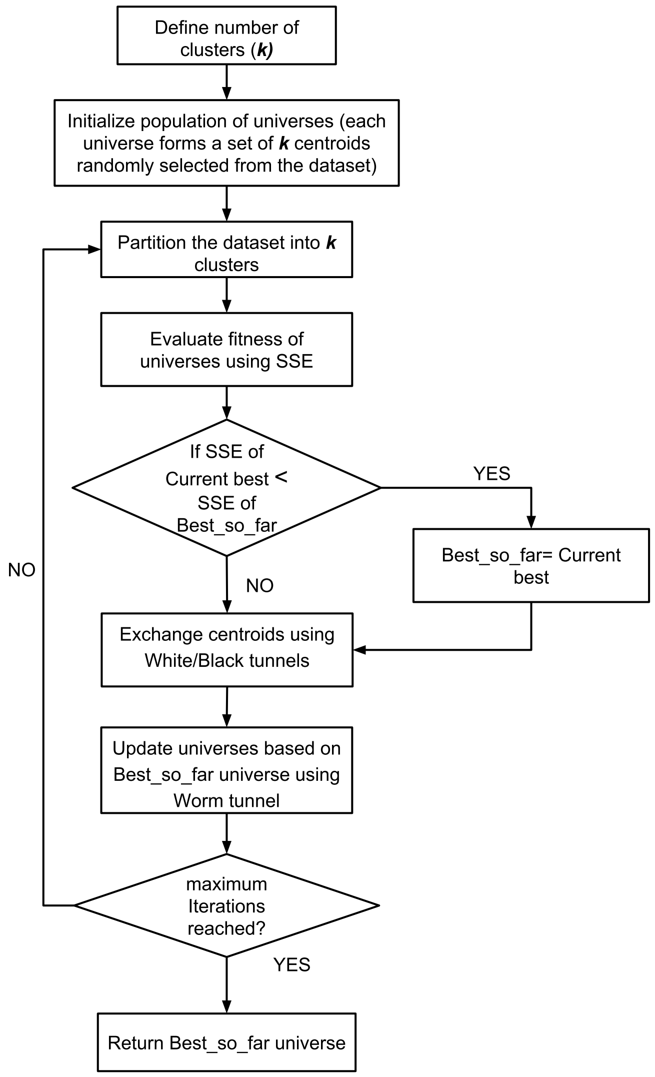

Define number of clusters (*k*)
Initialize population of universes (each universe forms a set of *k* centroids randomly selected from the dataset)
Partition the dataset into *k* clusters
Evaluate fitness of universes using SSE
If SSE of Current best < SSE of Best_so_far
YES
Best_so_far= Current best
NO
Exchange centroids using White/Black tunnels
Update universes based on Best_so_far universe using Worm tunnel
maximum Iterations reached?
NO
YES
Return Best_so_far universe

Fig. 2. Main processes of SCMVO algorithm.

7.1. Experiments environment and setup

For the experiments, Matlab software version 9.1 is used to implement the proposed SCMVO, DCMVO, and other nature-inspired based algorithms (CPSO and CGA). The experiments of HC and FF algorithms are conducted using Weka software version 3.8 (Hall et al., 2009). All experiments are conducted on a PC with Windows 7 Professional 64 bit operating System, an Intel(R) Core(TM) i7-6500U CPU with 16 GB RAM memory.

The number of iterations for SCMVO, CPSO and CGA is set to 150. After several initial experiments, we observed that this number suffices for all algorithms to converge. The umber of individuals is set to 50 for all optimizers. For SCMVO, P, min and max are parameters used in the update procedures. These parameters are selected after conducting extensive experiments. The other parameters for CGA and CPSO are set as per the recommendations of well-regarded published works. The parameters of the proposed SCMVO/DCMVO algorithms, CPSO, and CGA are set as listed in Table 1. The selection of MVO parameters was done after the extended experiments performed in Mirjalili et al. (2016). The lower bound *lb* and upper bound *ub* are set to 0 and 1, respectively since the datasets are normalized.

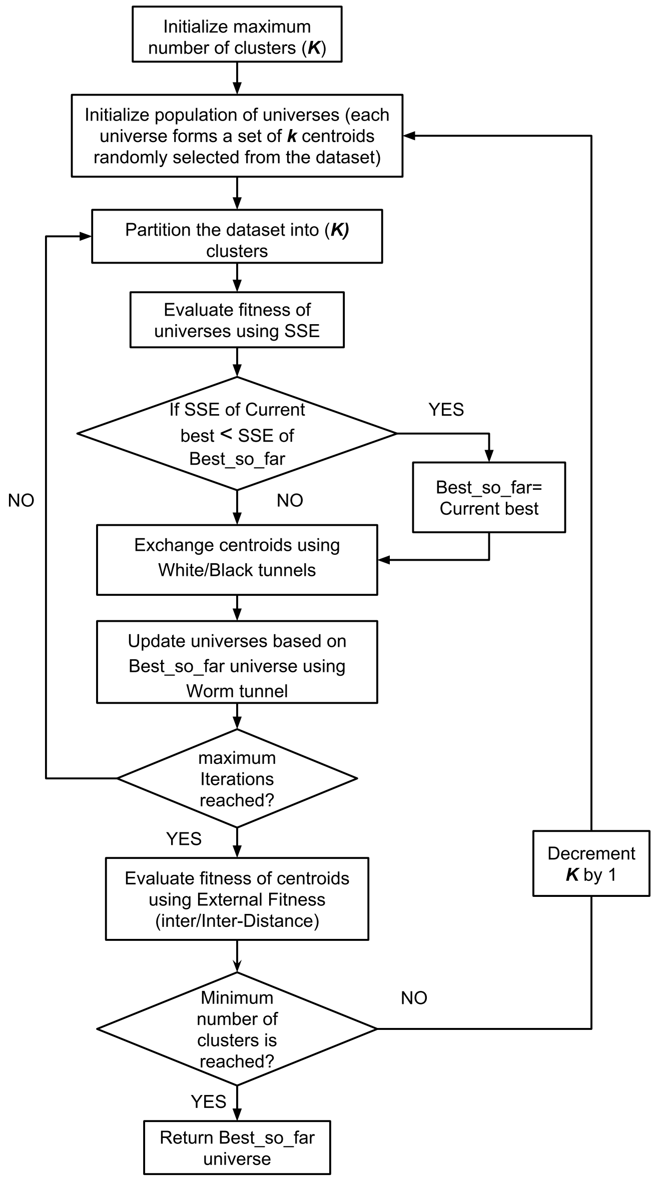

Initialize maximum number of clusters (*K*)
Initialize population of universes (each universe forms a set of *k* centroids randomly selected from the dataset)
Partition the dataset into (*K*) clusters
Evaluate fitness of universes using SSE
If SSE of Current best < SSE of Best_so_far
YES
Best_so_far= Current best
NO
NO
Exchange centroids using White/Black tunnels
Update universes based on Best_so_far universe using Worm tunnel
maximum Iterations reached?
YES
Evaluate fitness of centroids using External Fitness (inter/Inter-Distance)
Decrement *K* by 1
Minimum number of clusters is reached?
NO
YES
Return Best_so_far universe

Fig. 3. Main processes of DCMVO algorithm.

Table 1
Parameter settings.

Algorithm	Parameter	Value
SCMVO/DCMVO	Universe Size	50
	Min. existence probability	0.2
	Max. existence probability	1
CGA	Population Size	50
	Selection mechanism	Roulette wheel
	Cross-over	0.9
	Mutation	0.2
	Population size	50
CPSO	Acceleration constants	[2.1,2.1]
	Inertia weights	[0.9,0.6]
	Swarm size	50

In order to obtain statistically meaningful results from the non-deterministic clustering approaches (SCMVO, CPSO, CGA and K-Means) are run 30 times independently, then the average of results and the standard deviation are reported. The deterministic algorithms including HC and FF are applied only one time. In addition, all data sets are normalized to make all features in the same range [0,1].

Table 2
List of used datasets.

Dataset	No. of classes	No. of features	No. of data points	No. of objects in each cluster	Dataset type
Iris	3	4	150	50-50-50	Real
Glass	6	9	214	70-76-17-13-9-29	Real
Balance	3	4	625	49-288-288	Real
Hepatitis	2	10	155	32-123	Real
Breast cancer	2	8	699	458-241	Real
Wine	3	13	178	59-71-48	Real
Vertebral	2	6	310	207-100	Real
Ecoli	5	7	327	143-77-35-20-52	Real
Blood	2	4	748	570-178	Real
Seed	3	7	210	68-69-70	Real
Mouse	3	2	490	290-100-100	Artificial
Vary Density	3	2	150	50-50-50	Artificial

Table 3
Purity results of the SCMVO compared to other algorithms.

Dataset	SCMVO	*K*-Means	HC	FF	CPSO	CGA
	AVE ± STD	AVE ± STD	AVE ± STD	AVE ± STD	AVE ± STD	AVE ± STD
Iris	**0.96** ± **0.00**	0.57 ± 0.24	0.89 ± 0.00	0.86 ± 0.00	**0.96** ± **0.00**	**0.96** ± **0.00**
Glass	**0.52** ± **0.08**	0.36 ± 0.08	0.46 ± 0.00	0.48 ± 0.00	0.45 ± 0.00	0.36 ± 0.00
Balance	0.49 ± 0.17	0.47 ± 0.17	0.63 ± 0.00	**0.65** ± 0.00	0.37 ± 0.18	0.41 ± 0.18
Hepatitis	0.76 ± 0.04	0.71 ± 0.11	**0.79** ± **0.00**	**0.79** ± 0.00	0.76 ± 0.05	0.76 ± 0.04
Breast cancer	**0.96** ± **0.00**	0.95 ± 0.00	0.66 ± 0.00	0.84 ± 0.00	**0.96** ± 0.00	**0.96** ± 0.00
Wine	**0.96** ± **0.00**	0.90 ± 0.14	0.40 ± 0.00	0.70 ± 0.00	0.91 ± 0.12	0.95 ± 0.01
Vertebral	**0.71** ± **0.00**	0.68 ± 0.004	0.67 ± 0.00	0.68 ± 0.00	0.68 ± 0.02	0.70 ± 0.00
Ecoli	0.57 ± 0.13	**0.66** ± **0.11**	0.65 ± 0.00	0.60 ± 0.00	0.60 ± 0.11	0.62 ± 0.13
Blood	0.48 ± 0.01	0.53 ± 0.06	**0.76** ± **0.00**	**0.76** ± **0.00**	0.48 ± 0.00	0.48 ± 0.00
Seed	**0.90** ± 0.00	0.79 ± 0.19	**0.90** ± 0.00	0.67 ± 0.00	0.89 ± 0.05	**0.90** ± **0.00**
Mouse	0.86 ± 0.00	0.82 ± 0.07	**0.91** ± 0.00	0.80 ± 0.00	0.86 ± 0.00	0.86 ± 0.00
Vary Density	**0.95** ± **0.00**	0.66 ± 0.19	0.67 ± 0.00	0.67 ± 0.00	**0.95** ± **0.00**	**0.95** ± **0.00**

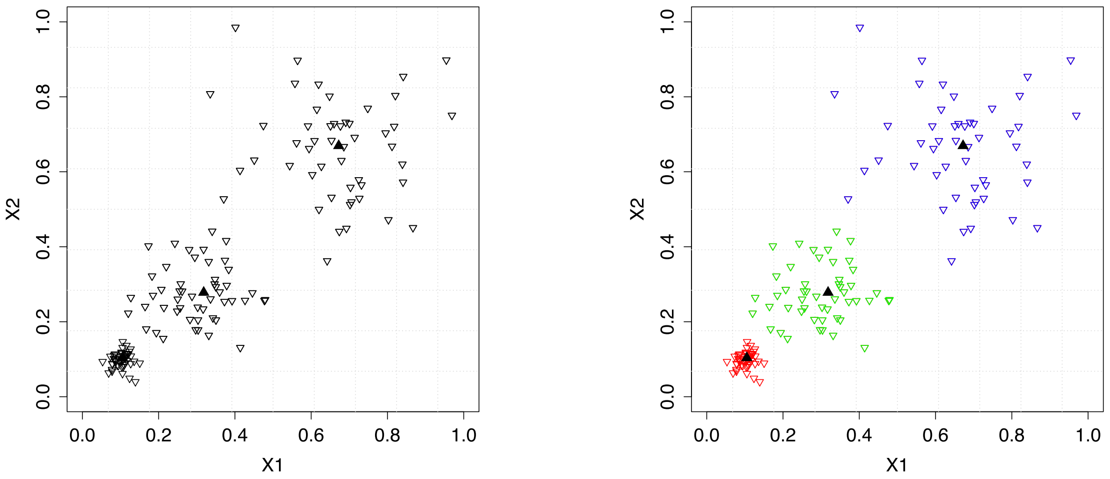
7.2. Datasets description

The performance of the proposed clustering approach will be assessed by applying it over twelve real and artificial datasets that are obtained from the UCI[1] (Blake and Merz, 1998) and ELKI[2] repositories. These datasets have a diverse number of features ranging from 2 to 13, and different number of classes ranging from 2 to 6. The details of all used datasets are given as follows:

- Iris: this dataset was created by Fisher (1936), which has been one of the popular pattern classification datasets in the literature. Iris dataset contains 3 classes and 150 instances, where each instance is expressed by three attributes: sepal length, sepal width, petal length, and petal width. Iris classes represent three types of iris plant: Setosa, Versicolor, and Virginica.
- Glass: this dataset was originally collected by B. German from Central Research Establishment in USA and used for crime identification (Evett and Spiehler, 1987). The dataset contains information about glass types and their contents in terms of oxide elements such as Sodium, Magnesium, Aluminum, Silicon, Potassium, Calcium, Barium, and Iron. The dataset contains 6 types of broken glasses which are exist in the crime scene. Some examples of glass types: building windows, vehicle windows, containers, tableware, and headlamps.
- Balance: this dataset was extracted by psychological experiments (Siegler, 1976) to model the balance scale into three cases; tip to the right, tip to the left, and balanced. The dataset consists of four attributes and 625 instances based on the balance direction, distance and weight.
- Hepatitis: this dataset contains 18 features measured for 155 patients affected by Hepatitis, which is a type of liver disease (Asuncion and Newman, 2007). The dataset was collected by Carnegie-Mellon University in USA. The dataset was used to predict the patient with Hepatitis disease if will be die or survive.
- Breast cancer: this dataset was originally obtained from the University of Wisconsin Hospitals, USA (Wolberg and Mangasarian, 1990). This dataset contains 699 instances where each instance is represented by 8 clinical tests. The class label of the dataset represents if the patient needs surgery for breast cancer or not.
- Wine: this dataset represents chemical analysis results of three types of wines grown in Italy but derived from different cultivars (Forina et al., 1991). The chemical analysis results produce 13 different constituents in each of the three types of wines. The objective of this dataset is to predict the type of wine based on these chemical results.
- Vertebral: this dataset includes 6 biomechanical features that used to classify orthopedic patients into two classes (normal or abnormal) (Berthonnaud et al., 2005). The attributes were derived from the shape and orientation of the pelvis and lumbar spine of the patient.
- Ecoli: the main purpose of this dataset is to detect the localization site of proteins by applying some measures about the cell such as cytoplasm, inner membrane, cleavable signal sequence and many others (Horton and Nakai, 1996). The dataset contains 7 attributes, and 327 instances.
- Blood: this dataset was created by Blood Transfusion Service Center, Taiwan, which contains information about blood donors (Yeh et al., 2009). The dataset contains four attributes: first donation, last donation, donation times, and volume of donation. This dataset contains 2 classes to represent whether the person makes donation or not.
- Seed: this dataset was created in the Institute of Agrophysics of the Polish Academy of Sciences in Lublin Charytanowicz et al. (2010). The dataset represents three different types of wheat (Kama, Rosa and Canadian). The dataset was built by high quality images of the internal kernel structure using X-ray technique by extracted 13 × 18 cm X-ray sections.
- Mouse and Vary Density datasets: these datasets are artificial data sets, which are generated using Gaussian distribution with variable densities (Schubert et al., 2015).

[1] http://archive.ics.uci.edu/ml/index.html.

[2] http://elki.dbs.ifi.lmu.de/wiki/DataSets.



(a) Original Data set.

(b) Clustering with SCMVO.

Fig. 4. Clustering results for the Vary Density dataset, where the black boxes represent the centroids (Purity = 0.95) (a) The original Vary Density dataset. (b) The clustering results with SCMVO.

Table 4
Entropy results of the SCMVO compared to other algorithms.

Dataset	SCMVO	*K*-Means	HC	FF	CGA	CPSO
	AVE ± STD	AVE ± STD	AVE ± STD	AVE ± STD	AVE ± STD	AVE ± STD
Iris	**0.14** ± 0.00	0.62 ± 0.35	0.23 ± 0.00	0.31 ± 0.00	**0.14** ± 0.00	**0.14** ± 0.00
Glass	0.22 ± 0.03	0.79 ± 0.10	0.66 ± 0.00	0.65 ± 0.00	0.60 ± 0.03	**0.17** ± 0.004
Balance	**0.61** ± 0.17	0.65 ± 0.16	0.74 ± 0.00	0.65 ± 0.00	0.68 ± 0.16	0.70 ± 0.16
Hepatitis	**0.28** ± 0.001	0.35 ± 0.15	0.73 ± 0.00	0.69 ± 0.00	**0.28** ± 0.00	**0.28** ± 0.00
Breast cancer	**0.15** ± 0.00	0.37 ± 0.17	0.93 ± 0.00	0.59 ± 0.00	**0.15** ± 0.00	**0.15** ± 0.00
Wine	0.58 ± 0.02	0.60 ± 0.04	0.98 ± 0.00	**0.53 ± 0.00**	0.57 ± 0.02	0.58 ± 0.03
Vertebral	**0.26** ± 0.02	0.27 ± 0.00	0.91 ± 0.00	0.91 ± 0.00	**0.26** ± 0.00	**0.26** ± 0.00
Ecoli	**0.13** ± 0.02	0.22 ± 0.18	0.52 ± 0.00	0.61 ± 0.00	0.17 ± 0.03	0.19 ± 0.16
Blood	**0.67** ± 0.01	0.86 ± 0.01	0.79 ± 0.00	0.78 ± 0.00	0.68 ± 0.01	0.73 ± 0.10
Seed	0.32 ± 0.02	0.36 ± 0.15	**0.30** ± 0.00	0.54 ± 0.00	0.32 ± 0.02	0.33 ± 0.02
Mouse	0.79 ± 0.00	0.79 ± 0.00	**0.17** ± 0.00	0.35 ± 0.00	0.79 ± 0.00	0.79 ± 0.00
Vary Density	0.32 ± 0.01	0.42 ± 0.21	0.42 ± 0.00	0.47 ± 0.00	**0.31** ± 0.01	0.33 ± 0.04

Table 5
P-values of the Wilcoxon test the purity results of SCMVO to *K*-Means and other nature-inspired algorithms (*P*-values ≤ 0.05 are bold).

Dataset	*K*-Means	CPSO	CGA
Iris	**4.17E−10**	1	1
Glass	**6.55E−08**	**0.000185**	**2.32E−11**
Balance	0.424456	**0.007734**	**0.044259**
Hepatitis	**0.006244**	1	1
Breast cancer	**2.71E−14**	0.333711	0.333711
Wine	**1.81E−08**	0.307597	**9.67E−07**
Vertebral	0.593385	0.177221	**6.71E−08**
Ecoli	0.00431	0.178176	0.074734
Blood	9.77E−13	0.693785	2.73E−06
Seed	0.160578	0.345589	0.060146
Mouse	**9.48E−14**	0.333711	0.333711
Vary Density	**1.64E−08**	1	1

Table 2 shows the summary of the datasets in terms of number of classes, number of features, number of data points, and data set type.

7.3. Static clustering results

Table 3 shows the average purity results for all clustering approaches. Note that high purity values indicate better results in terms of clustering quality.

As shown in Table 3, SCMVO outperformed *K*-Means, HC, FF, CPSO and CGA in three datasets (Glass, Wine and Vertebral). In addition, SCMVO has achieved the best performance for Iris, Breast cancer, Seed and Vary Density datasets along with SCPSO and CGA. On the other hand, *K*-Means outperformed SCMVO, HC, FF, CPSO and CGA in Ecoli dataset. HC has the best performance for Seed and Mouse datasets. However, its results for these datasets are very close to the SCMVO algorithm. Also, HC outperformed SCMVO, *K*-Means, CGA and CPSO and had an equal performance with FF in Hepatitis and Blood datasets. The FF algorithm achieved best results in three datasets: Balance, Hepatitis and Blood. However, the results of HC and FF are noticeably better than SCMVO and all other algorithms in Blood dataset, which might be due to the nature of this dataset. Finally, it can been seen that the best purity results obtained by CPSO and CGA were achieved by SCMVO as well. This means that the SCMVO clustering algorithm outperforms CPSO and CGA in all datasets except Ecoli dataset.

Overall, the results showed that the SCMVO provides the best performance in seven datasets. Also, SCMVO algorithm outperformed other algorithms in different datasets with diverse properties (*e.g.* the Vertebral dataset with two clusters). On the other hand, the Glass dataset has a higher number of clusters. For the rest of datasets, in which SCMVO did not outperformed the other algorithms, SCMVO was very competitive except the Blood dataset. It is also noticeable that the SCMVO algorithm shows low standard deviation rates, which indicates that the algorithm has good stability and robustness level. Fig. 4 shows an example of clustering process on the Vary Density dataset with two attributes.

All clustering algorithms are also evaluated using the entropy measurement. Table 4 shows the mean and standard deviation results of entropy for the 30 independent runs for SCMVO, *K*-Means, CPSO and

Table 6
The average number of clusters found by DCMVO and their corresponding error rates.

Dataset	Actual no. of clusters	DCMVO		DPSO		DGA	
		Auto. clusters	Error rate	Auto. clusters	Error rate	Auto. clusters	Error rate
Iris	3	2.97 ± 0.18	0.03 ± 0.18	3.03 ± 0.18	0.03 ± 0.18	3.07 ± 0.25	0.70 ± 0.25
Glass	6	5.60 ± 1.43	1.10 ± 0.91	2.00 ± 0.00	4.00 ± 0.00	2.00 ± 0.00	4.00 ± 0.00
Balance	3	2.00 ± 0.00	1.00 ± 0.00	2.00 ± 0.00	1.00 ± 0.00	2.00 ± 0.00	1.00 ± 0.00
Hepatitis	2	2.00 ± 0.00	0.00 ± 0.00	2.00 ± 0.00	0.00 ± 0.00	2.00 ± 0.00	0.00 ± 0.00
Breast cancer	2	2.00 ± 0.00	0.00 ± 0.00	2.00 ± 0.00	0.00 ± 0.00	2.00 ± 0.00	0.00 ± 0.00
Wine	3	2.00 ± 0.00	1.00 ± 0.00	2.00 ± 0.00	1.00 ± 0.00	2.00 ± 0.00	1.00 ± 0.00
Vertebral	2	2.00 ± 0.00	0.00 ± 0.00	2.20 ± 0.55	0.20 ± 0.55	2.00 ± 0.00	0.00 ± 0.00
Ecoli	5	2.00 ± 0.00	3.00 ± 0.00	2.00 ± 0.00	3.00 ± 0.00	2.00 ± 0.00	3.00 ± 0.00
Blood	2	2.00 ± 0.00	0.00 ± 0.00	2.00 ± 0.00	0.00 ± 0.00	2.03 ± 0.18	0.03 ± 0.18
Seed	3	2.00 ± 0.00	1.00 ± 0.00	2.00 ± 0.00	1.00 ± 0.00	2.30 ± 0.47	0.70 ± 0.47
Mouse	3	2.00 ± 0.00	1.00 ± 0.00	2.00 ± 0.00	1.00 ± 0.00	2.00 ± 0.00	1.00 ± 0.00
Vary Density	3	3.30 ± 0.48	0.08 ± 0.45	3.30 ± 0.18	0.08 ± 0.18	3.07 ± 0.25	0.07 ± 0.25

Table 7
Purity results of the DCMVO compared to other nature-inspired based clustering algorithms (DPSO and DGA).

Dataset	DCMVO	DPSO	DGA
	AVE ± STD	AVE ± STD	AVE ± STD
Iris	**0.96 ± 0.000**	0.95 ± 0.055	0.94 ± 0.058
Glass	**0.76 ± 0.055**	0.44 ± 0.000	0.44 ± 0.000
Balance	0.57 ± 0.140	**0.64 ± 0.168**	0.56 ± 0.135
Breast cancer	**0.96 ± 0.001**	**0.96 ± 0.000**	**0.96 ± 0.000**
Wine	0.60 ± 0.000	0.60 ± 0.000	**0.61 ± 0.023**
Hepatitis	**0.79 ± 0.000**	**0.79 ± 0.000**	**0.79 ± 0.000**
Vertebral	**0.71 ± 0.011**	0.67 ± 0.105	0.69 ± 0.008
Ecoli	**0.64 ± 0.001**	**0.64 ± 0.000**	**0.64 ± 0.000**
Blood	**0.76 ± 0.000**	**0.76 ± 0.000**	**0.76 ± 0.033**
Seed	0.66 ± 0.000	0.67 ± 0.000	**0.74 ± 0.108**
Mouse	**0.73 ± 0.019**	0.67 ± 0.106	0.72 ± 0.022
Vary Density	0.91 ± 0.078	**0.95 ± 0.024**	0.94 ± 0.033

Table 8
Entropy results of the DCMVO compared to other nature-inspired based clustering algorithms (DPSO and DGA).

Dataset	DCMVO	DPSO	DGA
	AVE ± STD	AVE ± STD	AVE ± STD
Iris	**0.14 ± 0.00**	0.13 ± 0.008	0.13 ± 0.027
Glass	**0.23 ± 0.073**	0.68 ± 0.001	0.68 ± 0.000
Balance	0.75 ± 0.170	**0.69 ± 0.196**	0.76 ± 0.148
Hepatitis	**0.57 ± 0.019**	**0.57 ± 0.014**	**0.57 ± 0.020**
Breast cancer	**0.26 ± 0.002**	**0.26 ± 0.000**	**0.26 ± 0.000**
Wine	0.60 ± 0.002	0.60 ± 0.001	**0.59 ± 0.023**
Vertebral	0.68 ± 0.008	**0.63 ± 0.136**	0.69 ± 0.007
Ecoli	**0.57 ± 0.002**	**0.57 ± 0.000**	**0.57 ± 0.000**
Blood	0.79 ± 0.000	0.79 ± 0.000	**0.78 ± 0.037**
Seed	0.53 ± 0.001	0.53 ± 0.000	0.46 ± 0.104
Mouse	**0.53 ± 0.016**	0.63 ± 0.136	0.54 ± 0.018
Vary Density	**0.12 ± 0.039**	0.14 ± 0.017	0.14 ± 0.024

CGA. In contrast to the results obtained using the purity measure, less entropy values indicates better result in terms of the clustering quality.

As shown in Table 4, SCMVO outperformed other algorithms in three datasets (Balance, Ecoli and Blood), and it has best performance as CPSO and CGA in Iris, Hepatitis, Breast cancer and Vertebral datasets. Each of the CGA and CPSO outperformed the rest of algorithms in only one dataset. However, their results are not very different than those obtained by SCMVO. HC provides best results in two datasets, whereas FF shows the best results in only one dataset. Note that K-Means did not achieve the best results in any of the algorithms. As a conclusion, SCMVO algorithm showed good capabilities in clustering the benchmarking datasets in terms of entropy measurement and achieved the best or very competitive results in most of the datasets.

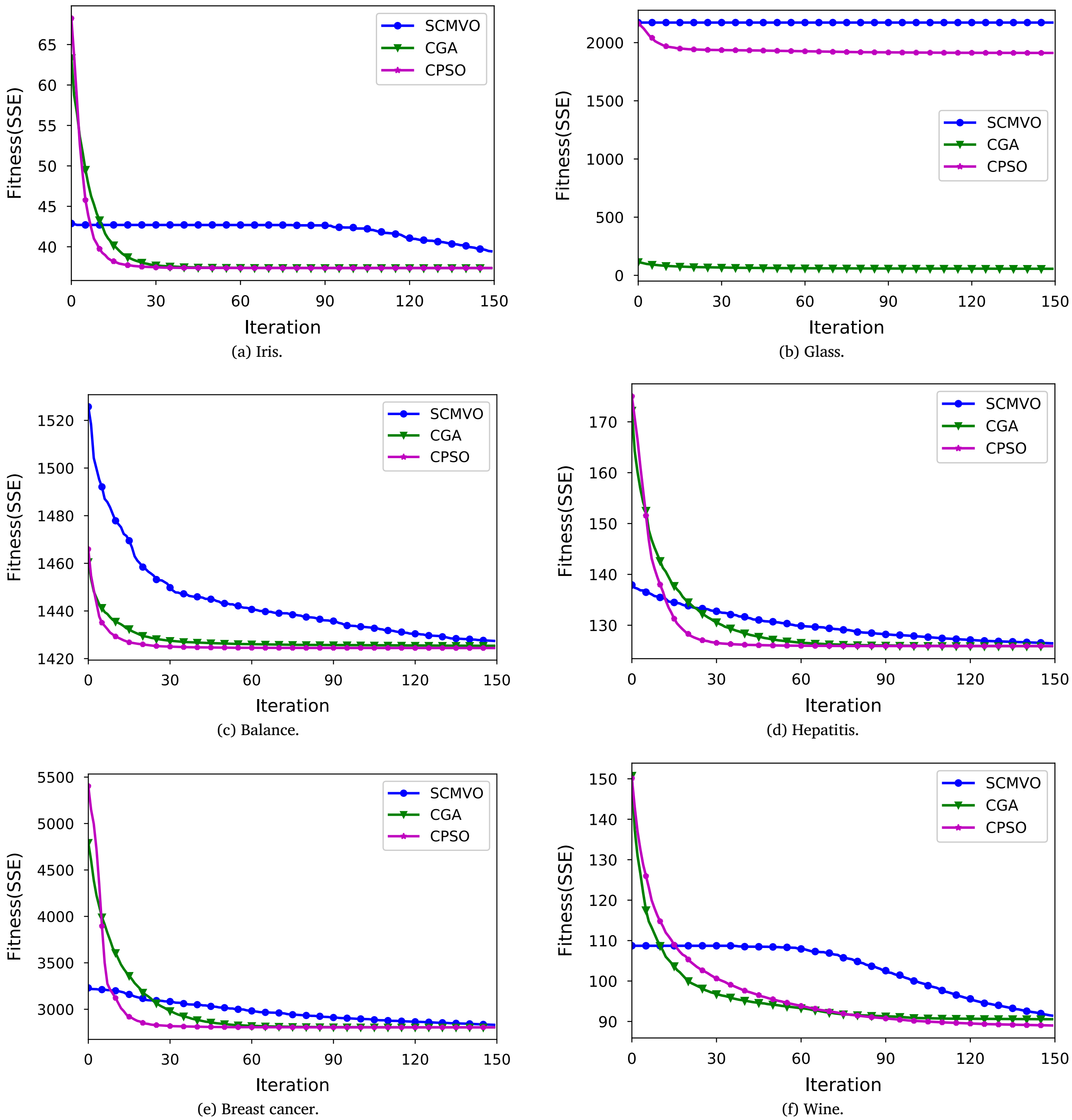
The average convergence curves for SCMVO, CPSO and CGA are shown in Figs. 5 and 6. The purpose of drawing the convergence curves is to observe how an algorithms approaches the global optimum starting from a random population of solutions.

The convergence curves of the datasets shows that SCMVO is the fastest in two datasets (Glass and Vertebral), while CPSO ranks first only in the Mouse dataset. Moreover, K-Means shows the fastest convergence rate when solving resting datasets (Iris, Seed, Vary Density, Balance, Blood, Breast cancer, Ecoli, Wine and Hepatitis). SCMVO was the slowest algorithm in ten datasets. The curves for this algorithm follow the same pattern: the convergence rate is low in the initial steps of optimization and show accelerated behavior proportional to the number of iterations. On the other hand, the K-Means algorithm shows fast convergence rates, but there is no acceleration, which results in low purity and high entropy results. This due to the main drawback of the K-Means algorithm which is the high probability to trap in the local optimum. As a summary, although the SCMVO algorithm does not shows the fast convergence, it provides better results than K-Means, CGA, CPSO, HC, and FF in terms of purity and entropy.

In order to see whether the obtained results of SCMVO are statistically significant or not, the nonparametric Wilcoxon rank-sum test is

Table 9
Purity and entropy results of the SCMVO compared to DCMVO.

Dataset	Purity		Entropy	
	SCMVO	DCMVO	SCMVO	DCMVO
	AVE ± STD	AVE ± STD	AVE ± STD	AVE ± STD
Iris	**0.96 ± 0.00**	**0.96 ± 0.00**	**0.14 ± 0.00**	**0.14 ± 0.00**
Glass	0.52 ± 0.08	**0.76 ± 0.055**	**0.22 ± 0.03**	0.23 ± 0.073
Balance	0.49 ± 0.17	**0.57 ± 0.14**	**0.61 ± 0.17**	0.75 ± 0.17
Hepatitis	0.76 ± 0.04	**0.79 ± 0.00**	**0.28 ± 0.001**	0.57 ± 0.019
Breast cancer	**0.96 ± 0.00**	**0.96 ± 0.001**	**0.15 ± 0.00**	0.26 ± 0.002
Wine	**0.96 ± 0.00**	0.60 ± 0.00	**0.58 ± 0.02**	0.6 ± 0.002
Vertebral	**0.71 ± 0.00**	**0.71 ± 0.011**	**0.26 ± 0.02**	0.68 ± 0.008
Ecoli	0.57 ± 0.13	**0.64 ± 0.001**	**0.13 ± 0.02**	0.57 ± 0.002
Blood	0.48 ± 0.01	**0.76 ± 0.00**	**0.67 ± 0.01**	0.79 ± 0.00
Seed	**0.90 ± 0.00**	0.66 ± 0.00	**0.32 ± 0.02**	0.53 ± 0.001
Mouse	**0.86 ± 0.00**	0.73 ± 0.019	0.79 ± 0.00	**0.53 ± 0.016**
Vary Density	**0.95 ± 0.00**	0.91 ± 0.078	0.32 ± 0.01	**0.12 ± 0.039**



(a) Iris.

(b) Glass.

(c) Balance.

(d) Hepatitis.

(e) Breast cancer.

(f) Wine.

Fig. 5. Convergence curves of SCMVO, CGA, and CPSO for Iris, Glass, Balance, Hepatitis, Breast cancer, and Wine datasets.

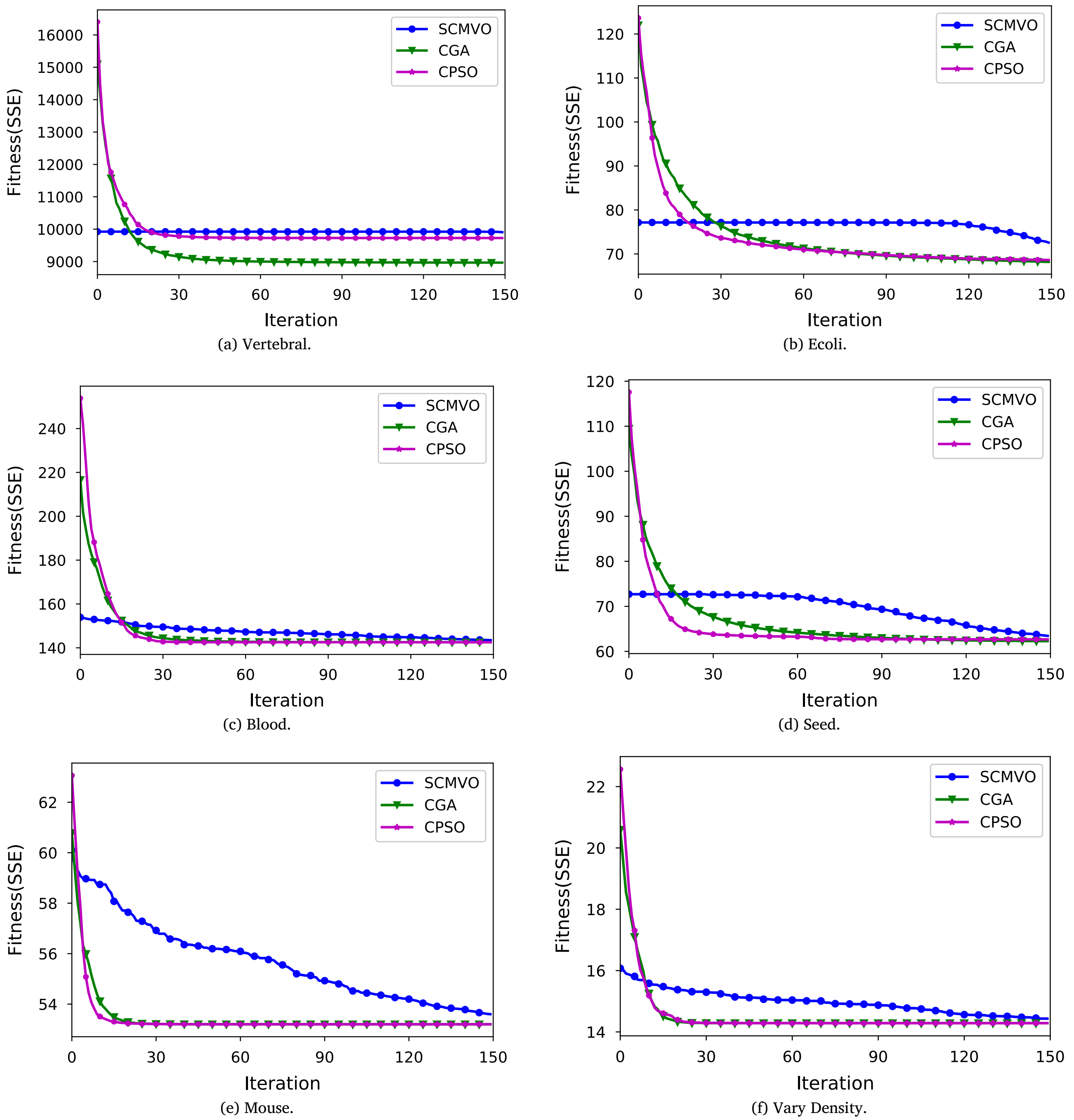
used at 5% significance level. The test is applied for the SCMVO against the other nondeterministic clustering approaches: K-Means, CPSO and CGA. Table 5 shows the obtained p-values of the Wilcoxon rank-sum test.

Inspecting the results in Table 5, SCMVO is significantly better than the K-Means algorithm in seven datasets (Iris, Glass, Hepatitis, Breast cancer, Wine, Mouse and Vary Density), while SCMVO was significantly better than CPSO in two datasets (Glass and Balance). In addition, the SCMVO algorithm is substantially better than CGA in four datasets (Glass, Balance, Wine and Vertebral). This is an evidence of the significance of the SCMVO results when compared to K-Means, SCPSO and CGA.

7.4. Dynamic clustering results

The second phase of the experiments is to evaluate and benchmark the proposed dynamic CMVO (DCMVO) algorithm on the datasets listed previously in Table 2. The task of DCMVO is to automatically find the best number of centroids along with their values.

The best number of centroids found by DCMVO is validated by calculating the error measurement given in Eq. (11), which equals to the difference between the actual number of clusters k and the number of clusters automatically found by DCMVO. Table 6 shows the average error and its standard deviation calculated over 30 runs for each dataset. Figs. 7 and 8 show the convergence curves of DCMVO, DPSO, and DGA in terms of external fitness value. The value on the X-axis (number of the



(a) Vertebral.

(b) Ecoli.

(c) Blood.

(d) Seed.

(e) Mouse.

(f) Vary Density.

Fig. 6. Convergence curves of SCMVO, CGA, and CPSO for Vertebral, Ecoli, Blood, Seed, Mouse, and Vary Density datasets.

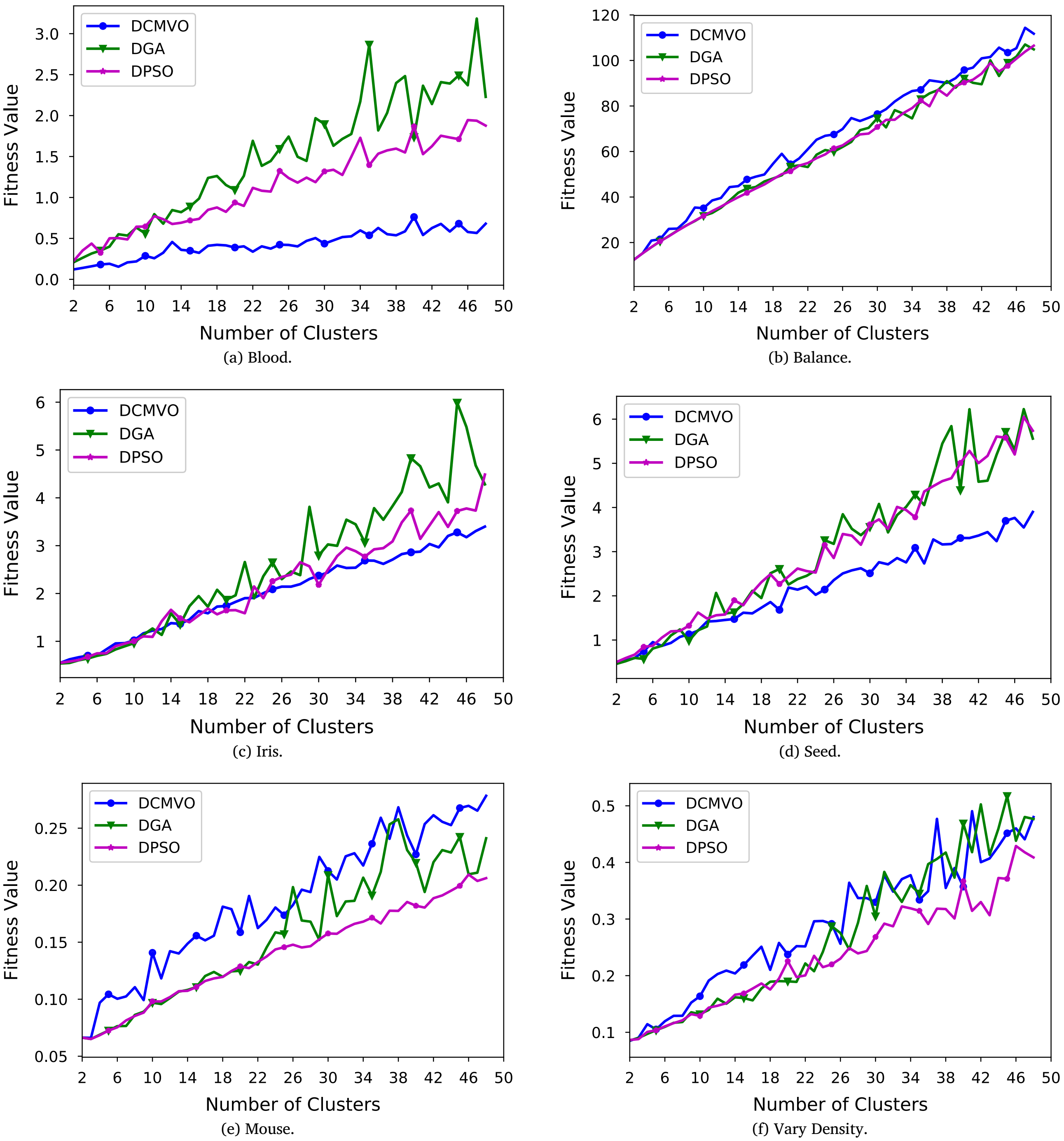
predicted clusters) that has the lowest external fitness value on the Y-axis in each curve indicates the best number of clusters in each dataset.

$$\text{Error} = \left| K - K_{\text{predicted}} \right| \tag{11}$$

As per the results listed in Table 6, it is evident that DCMVO automatically determines the number of clusters in each dataset in a way that is very close the actual number of clusters for most of the cases. In four datasets including Blood, Breast cancer, Hepatitis and Vertebral the exact number of clusters (i.e. 2) is detected in the 30 runs, therefore the standard deviation is 0. In addition, the DCMVO managed to accurately predict the actual number of clusters for Iris, Glass and Vary Density datasets with a slight marginal error. DCMVO, DPSO and DGA were very competitive in finding the optimal number of clusters in all datasets except for Glass dataset, where DPSO and DGA detected only 2 clusters while DCMVO detected 5.6 on average which is very close to the ground truth (i.e. 6 clusters).

The results of the purity and entropy measures are shown in Tables 7 and 8, respectively. In terms of purity, it can be observed that DCMVO obtained better results than DPSO in four datasets, whereas they have same purity results in five datasets. On the other hand, DCMVO achieved better purity ratios than DGA in five datasets and they have similar ratios in four datasets.

Taken together, the MVO algorithm performed well in both static and dynamic clustering approaches proposed in this work. Clustering problems have a highly multi-modal search space and require an algorithm with suitable mechanisms to avoid them all. The MVO algorithm constantly combines solutions during optimization that results in an extremely high exploration. This assisted MVO to perform



(a) Blood.

(b) Balance.

(c) Iris.

(d) Seed.

(e) Mouse.

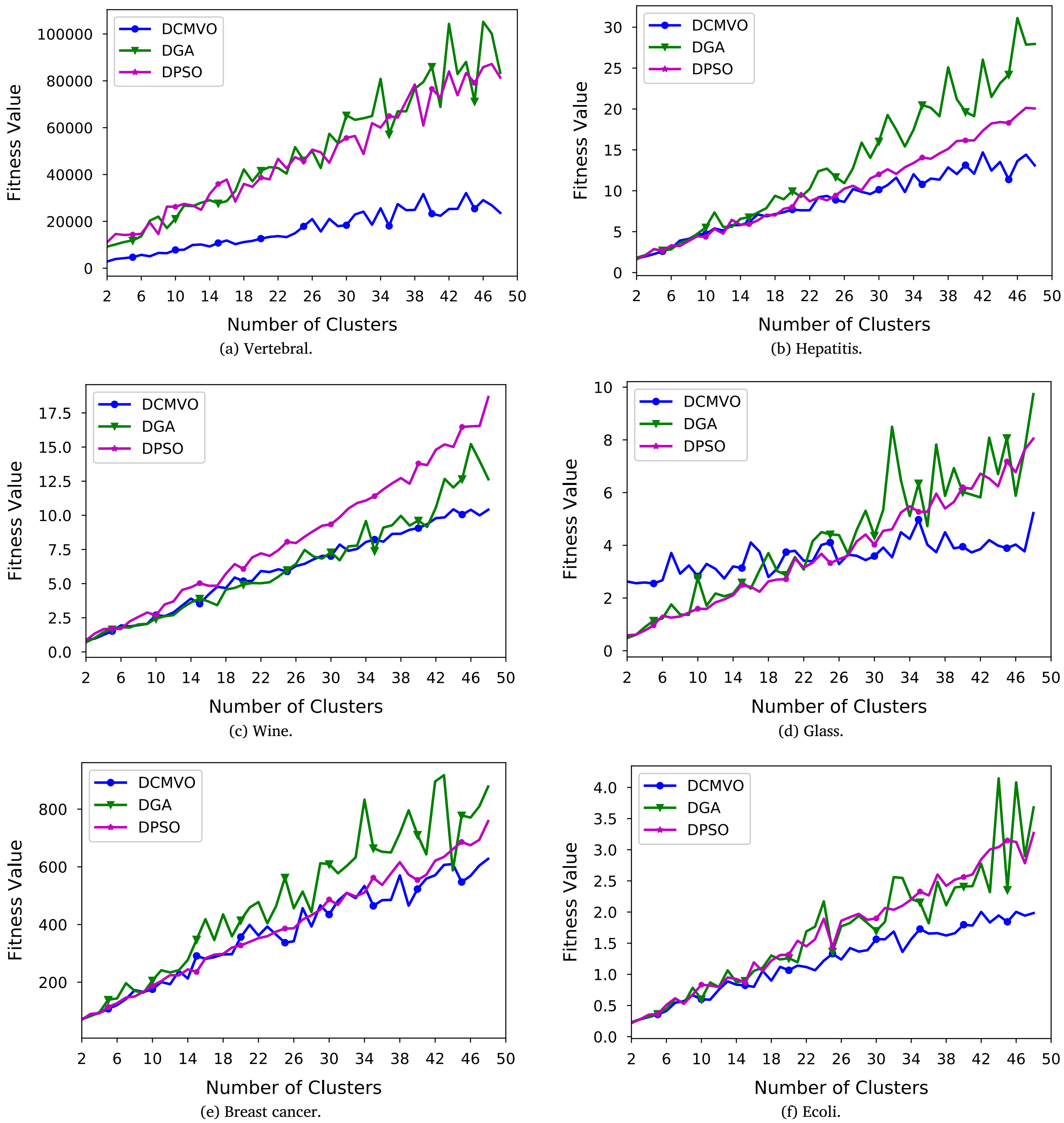
(f) Vary Density.

Fig. 7. Convergence curves of DCMVO, DPSO, and DGA for Blood, Balance, Iris, Seed, Mouse, and Vary Density datasets.

really well on the clustering problems employed in this work. High exploration normally results in poor quality global optimum though. This means that an algorithm needs to decrease exploration after finding a promising area to exploit it. The MVO algorithm benefits from two adaptive mechanisms in this regard, which are the main reason why it outperformed other algorithms in terms of the accuracy of results.

7.5. Comparing DCMVO to SCMVO

Table 9 presents the comparative results of SCMVO and DCMVO algorithms in terms of purity and entropy measures on 12 datasets. Inspecting the results quantified by the purity measure, it can be seen that the results of DCMVO are very competitive to SCMVO. DCMVO achieved the best purity results in 8 datasets, while SCMVO achieved the best results in seven 7 datasets. Considering the entropy results, although that CSMVO performed better in most of the datasets, DCMVO was superior in Mouse and Vary Density datasets and very competitive in other three datasets which are Iris, Glass and Wine. This slight superiority of SCMVO is due to the priori information about the true number of clusters. This information reduces the search space for SCMVO and the complexity of the problem. Overall, the results show that DCMVO is still very competitive specially when no priori information is available about the number of clusters in the data.



(a) Vertebral.

(b) Hepatitis.

(c) Wine.

(d) Glass.

(e) Breast cancer.

(f) Ecoli.

Fig. 8. Convergence curves of DCMVO, DPSO, and DGA for Vertebral, Hepatitis, Wine, Glass, Breast cancer, and Ecoli datasets.

8. Conclusions

In this work, a new evolutionary clustering algorithm was developed based on a recent nature-inspired metaheuristic algorithm called Multi-verse Optimizer (MVO). Two versions of the algorithm were proposed: a static version (SCMVO), and a dynamic version (DCMVO). In SCMVO, the number of clusters has to be defined prior the application. In contrast, the DCMVO automatically detects the number of clusters and solves one of the most common drawbacks of typical clustering algorithms. To validate the proposed clustering algorithms, they were employed to cluster twelve real and artificial benchmarks. The performance of algorithms were quantified using purity and entropy measures. The conducted experiments and the statistical analysis showed promising results for the SCMVO and DCMVO algorithms when compared to other traditional and evolutionary clustering algorithms.

For future work, more investigations of the proposed approaches can be added such as the verification of proposed algorithms on other types of data sets with larger number of dimensions. In addition, the proposed approaches can be applied on real world applications such as image segmentation and web logs clustering.

References

Agustı, L., Salcedo-Sanz, S., Jiménez-Fernández, S., Carro-Calvo, L., Del Ser, J., Portilla-Figueras, J.A., et al., 2012. A new grouping genetic algorithm for clustering problems. Expert Syst. Appl. 39 (10), 9695–9703.

Ahmadyfard, A., Modares, H., 2008. Combining pso and *K*-Means to enhance data clustering. In: Telecommunications, 2008. IST 2008. International Symposium on. IEEE, pp. 688–691.

Aljarah, I., Ludwig, S.A., 2012. Parallel particle swarm optimization clustering algorithm based on mapreduce methodology. In: Nature and Biologically Inspired Computing, NaBIC, 2012 Fourth World Congress on. IEEE, pp. 104–111.

Aljarah, I., Ludwig, S.A., 2013a. Mapreduce intrusion detection system based on a particle swarm optimization clustering algorithm. In: Evolutionary Computation, CEC, 2013 IEEE Congress on. IEEE, pp. 955–962.

Aljarah, I., Ludwig, S.A., 2013b. A new clustering approach based on glowworm swarm optimization. In: Evolutionary Computation, CEC, 2013 IEEE Congress on. IEEE, pp. 2642–2649.

Aljarah, I., Ludwig, S.A., 2013c. Towards a scalable intrusion detection system based on parallel PSO clustering using mapreduce. In: Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation. ACM, pp. 169–170.

Aljarah, I., Ala'M, A.Z., Faris, H., Hassonah, M.A., Mirjalili, S., Saadeh, H., 2018. Simultaneous Feature Selection and Support Vector Machine Optimization Using the Grasshopper Optimization Algorithm. Cognitive Computation, 1–18.

Al-Madi, N., Aljarah, I., Ludwig, S.A., 2014. Parallel glowworm swarm optimization clustering algorithm based on mapreduce. In: Swarm Intelligence, SIS, 2014 IEEE Symposium on. IEEE, pp. 1–8.

Asuncion, A., Newman, D., 2007. UCI machine learning repository.

Bandyopadhyay, S., Coyle, E.J., 2003. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In: INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, vol. 3. IEEE, pp. 1713–1723.

Baraldi, A., Blonda, P., 1999. A survey of fuzzy clustering algorithms for pattern recognition. i. IEEE Trans. Syst. Man Cybern. B 29 (6), 778–785.

Bellot, P., El-Bèze, M., 1999. A Clustering Method for Information Retrieval, Tech. Rep. Technical ReportIR-0199. Laboratoire d'Informatique d'Avignon, France.

Berthonnaud, E., Dimnet, J., Roussouly, P., Labelle, H., 2005. Analysis of the sagittal balance of the spine and pelvis using shape and orientation parameters. Clin. Spine Surg. 18 (1), 40–47.

Blake, C.L., Merz, C.J., 1998. UCI Repository of Machine Learning Databases. University of California, Department of Information and Computer Science, Irvine, CA.

Caliński, T., Harabasz, J., 1974. A dendrite method for cluster analysis. Comm. Statist. Theory Methods 3 (1), 1–27.

Charytanowicz, M., Niewczas, J., Kulczycki, P., Kowalski, P.A., Łukasik, S., Żak, S., 2010. Complete gradient clustering algorithm for features analysis of x-ray images. In: Information Technologies in Biomedicine. Springer, pp. 15–24.

Ding, Y., Fu, X., 2016. Kernel-based fuzzy c-means clustering algorithm based on genetic algorithm. Neurocomputing 188, 233–238.

Donalek, C., Supervised and unsupervised learning, in: Astronomy Colloquia, USA, 2011.

Doval, D., Mancoridis, S., Mitchell, B.S., 1999. Automatic clustering of software systems using a genetic algorithm. In: Software Technology and Engineering Practice, 1999. STEP'99. Proceedings. IEEE, pp. 73–81.

Evett, I.W., Spiehler, E., 1987. Rule induction in forensic science. KBS in Goverment, Online Publications, pp. 107–118.

Faris, H., Aljarah, I., Mirjalili, S., 2016. Training feedforward neural networks using multi-verse optimizer for binary classification problems. Appl. Intell. 45 (2), 322–332.

Faris, H., Hassonah, M.A., Ala'M, A.-Z., Mirjalili, S., Aljarah, I., 2017. A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture. Neural Comput. Appl. 1–15.

Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Ann. Hum. Genet. 7 (2), 179–188.

Forina, M., 1991. et al., An Extendible Package for Data Exploration, Classification and Correlation, Institute of Pharmaceutical and Food Analisys and Technologies, Via Brigata Salerno, 16147.

Fung, B.C., Wang, K., Ester, M., 2003. Hierarchical document clustering using frequent itemsets. In: Proceedings of the 2003 SIAM International Conference on Data Mining. SIAM, pp. 59–70.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2009. The weka data mining software: an update. ACM SIGKDD Explor. Newsl. 11 (1), 10–18.

Hassanzadeh, T., Meybodi, M.R., 2012. A new hybrid approach for data clustering using firefly algorithm and *K*-Means. In: Artificial Intelligence and Signal Processing, AISP, 2012 16th CSI International Symposium on. IEEE, pp. 007–011.

Hatamlou, A., 2013. Black hole: A new heuristic optimization approach for data clustering. Inform. Sci. 222, 175–184.

Hochbaum, D.S., Shmoys, D.B., 1985. A best possible heuristic for the k-center problem. Math. Oper. Res. 10 (2), 180–184.

Horton, P., Nakai, K., 1996. A probabilistic classification system for predicting the cellular localization sites of proteins. In: Ismb, vol. 4. pp. 109–115.

Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data clustering: a review. ACM Comput. Surv. 31 (3), 264–323.

Jangir, P., Parmar, S.A., Trivedi, I.N., Bhesdadiya, R., 2017. A novel hybrid particle swarm optimizer with multi verse optimizer for global numerical optimization and optimal reactive power dispatch problem. Eng. Sci. Technol. Int. J. 20 (2), 570–586.

Jarboui, B., Cheikh, M., Siarry, P., Rebai, A., 2007. Combinatorial particle swarm optimization (CPSO) for partitional clustering problem. Appl. Math. Comput. 192 (2), 337–345.

Jardine, N., van Rijsbergen, C.J., 1971. The use of hierarchic clustering in information retrieval. Inf. Storage Retr. 7 (5), 217–240.

Jianliang, M., Haikun, S., Ling, B., 2009. The application on intrusion detection based on *K*-Means cluster algorithm. In: Information Technology and Applications, 2009. IFITA'09. International Forum on, vol. 1. IEEE, pp. 150–152.

Karypis, G., Han, E.-H., Kumar, V., 1999. Chameleon: Hierarchical clustering using dynamic modeling. Computer 32 (8), 68–75.

Kumar, M., et al., 2013. An optimized farthest first clustering algorithm. In: Engineering (NUiCONE), 2013 Nirma University International Conference on. IEEE, pp. 1–5.

Lee, C.-Y., Antonsson, E., 2000. Dynamic partitional clustering using evolution strategies. In: Industrial Electronics Society, 2000. IECON 2000. 26th Annual Confjerence of the IEEE, vol. 4. IEEE, pp. 2716–2721.

MacQueen, J., et al., Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, Oakland, CA, USA, 1967, pp. 281–297.

Mafarja, M., Aljarah, I., Heidari, A.A., Hammouri, A.I., Faris, H., Ala'M, A.Z., Mirjalili, S., 2017. Evolutionary Population Dynamics and Grasshopper Optimization Approaches for Feature Selection Problems. Knowledge-Based Systems.

Marroquin, J.L., Girosi, F., 1993. Some Extensions of the *K*-Means Algorithm for Image Segmentation and Pattern Classification, Tech. Rep. MIT Al Lab., AIMemo 1390.

Maulik, U., Bandyopadhyay, S., 2000. Genetic algorithm-based clustering technique. Pattern Recognit. 33 (9), 1455–1465.

Mirjalili, S., Mirjalili, S.M., Hatamlou, A., 2016. Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput. Appl. 27 (2), 495–513.

Omran, M., Engelbrecht, A.P., Salman, A., 2005. Particle swarm optimization method for image clustering. Int. J. Pattern Recognit. Artif. Intell. 19 (03), 297–321.

Ozturk, C., Hancer, E., Karaboga, D., 2015. Dynamic clustering with improved binary artificial bee colony algorithm. Appl. Soft Comput. 28, 69–80.

Rao, A.S., Ramakrishna, S., Babu, P.C., 2016. MODC: Multi-objective distance based optimal document clustering by GA. Indian J. Sci. Technol. 9, 1–8.

Reddy, S., Parker, A., Hyman, J., Burke, J., Estrin, D., Hansen, M., 2007. Image browsing, processing, and clustering for participatory sensing: lessons from a dietsense prototype. In: Proceedings of the 4th Workshop on Embedded Networked Sensors. ACM, pp. 13–17.

Scheunders, P., 1997. A genetic c-means clustering algorithm applied to color image quantization. Pattern Recognit. 30 (6), 859–866.

Schubert, E., Koos, A., Emrich, T., Züfle, A., Schmid, K.A., Zimek, A., 2015. A framework for clustering uncertain data. Proc. VLDB Endowment 8 (12), 1976–1979.

Senthilnath, J., Omkar, S., Mani, V., 2011. Clustering using firefly algorithm: performance study. Swarm Evolut. Comput. 1 (3), 164–171.

Shelokar, P., Jayaraman, V.K., Kulkarni, B.D., 2004. An ant colony approach for clustering. Anal. Chim. Acta 509 (2), 187–195.

Siegler, R.S., 1976. Three aspects of cognitive development. Cogn. Psychol. 8 (4), 481–520.

Strehl, A., Ghosh, J., Mooney, R., Impact of similarity measures on web-page clustering, in: Workshop on Artificial Intelligence for Web Search (AAAI 2000), 58, 2000, p. 64.

Trivedi, I.N., Jangir, P., Jangir, N., Parmar, S.A., Bhoye, M., Kumar, A., 2016. Voltage stability enhancement and voltage deviation minimization using multi-verse optimizer algorithm. In: Circuit, Power and Computing Technologies, ICCPCT, 2016 International Conference on. IEEE, pp. 1–5.

Van der Merwe, D., Engelbrecht, A.P., 2003. Data clustering using particle swarm optimization. In: Evolutionary Computation, 2003. CEC'03. the 2003 Congress on, vol. 1. IEEE, pp. 215–220.

White, S.D., Frenk, C.S., 1991. Galaxy formation through hierarchical clustering. Astrophys. J. 379, 52–79.

Wolberg, W.H., Mangasarian, O.L., 1990. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. Proc. Natl. Acad. Sci. 87 (23), 9193–9196.

Xie, Y., Phoha, V.V., 2001. Web user clustering from access log using belief function. In: Proceedings of the 1st International Conference on Knowledge Capture. ACM, pp. 202–208.

Xu, R., Wunsch, D., 2005. Survey of clustering algorithms. IEEE Trans. Neural Netw. 16 (3), 645–678.

Yang, X.-S., 2011. Review of meta-heuristics and generalised evolutionary walk algorithm. Int. J. Bio-Inspired Comput. 3 (2), 77–84.

Yeh, I.-C., Yang, K.-J., Ting, T.-M., 2009. Knowledge discovery on RFM model using Bernoulli sequence. Expert Syst. Appl. 36 (3), 5866–5871.