Astrophysics inspired multi-objective approach for automatic clustering and feature selection in real-life environment

Gaurav Dhiman* and Vijay Kumar†

Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, Punjab 147004, India

* gaurav.dhiman@thapar.edu
† vijay.kumar@thapar.edu

Received 12 July 2018
Revised 3 September 2018
Accepted 20 September 2018
Published 1 November 2018

In this paper, a novel astrophysics-based approach is proposed for automatically finding the clusters and features simultaneously. A novel agent encoding scheme is used to encode both the number of clusters and features. A novel dynamic threshold technique is proposed for an efficient searching. The validation of proposed technique is tested on eight real-life data sets. The statistical significance of proposed technique is attributed by statistical tests. It is also applied on solving image segmentation and microarray data analysis problems. Experimental results reveal that the proposed technique outperforms the other competitive approaches.

Keywords: Astrophysics; clustering; feature selection; multi-objective optimization.

1. Introduction

Clustering is an un-supervised machine learning approach which is used to partition the data set into relevant clusters. There may be some criteria and rules to partition the data set. These rules may be intrinsic or extrinsic based on the problem under consideration. Clustering is widely used in various research domains such as machine learning, pattern recognition analysis, optimization, image segmentation and many more.¹⁻⁴ Clustering algorithms are divided into two main categories such as hierarchical and partitional. Hierarchical clustering consists of hierarchical nesting of each clustering level.⁵ These suffer from overlapping clusters and do not support dynamism.⁶,⁷ The latter decomposes the data set into disjoint sets. Partitional clustering algorithm is more reliable as compared to hierarchical clustering.⁸ Hence, the main focus of this paper is partition-based clustering algorithm. Partitional clustering algorithms consider equal weightage for all the features. This may
reduce the performance of clustering problem. Another challenging task in partitional clustering is to find an optimal number of clusters. Feature selection can be used to reduce the redundant and irrelevant features to improve the clustering accuracy and performance. There is a need to develop an automatic clustering approach to find relevant features and clusters simultaneously during the run for solving real-life problems.

In this paper, a novel approach named astrophysics-based multi-objective spotted hyena optimizer for automatic clustering and feature selection (AMOSHO, CFS) is used. A composite agent representation is designed for encoding both clusters and features separately. A new threshold concept is also proposed to figure out the clusters that are optimal and select features accurately. The threshold values for a given data set is computed on the basis of variance. The performance of proposed approach is evaluated using three clustering metrics such as the number of clusters, relevant features, and classification error. The performance of the proposed approach is compared with four state-of-the-art algorithms namely FeaClusMOO, FeaClusMOO_EUC, variable string length point symmetry-based clustering technique (VGAPS), and K-means.

The rest of the paper is structured as follows. Section 2 briefly describes the background of clustering problem and multi-objective spotted hyena optimizer (MOSHO). Section 3 presents the existing work done in the field of multi-objective clustering and feature selection techniques. Section 4 discusses the proposed approach. Section 5 covers the experimental results and discussions. In Sec. 6, the proposed approach is tested on two real-life applications. Finally, conclusion is presented in Sec. 7.

2. Background

This section discusses the basic concepts of data clustering and MOSHO.

2.1. Data clustering

The clustering algorithm aims to minimize the within cluster variation and maximize the between cluster variation. The mathematical formulation of partitional clustering is described as follows.

Let \( X = \{x_1, x_2, \ldots, x_n\} \) be the set of \( n \) data points, where \( x_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\} \in R^k \). Each measure \( x_{ij} \) is a feature. Clustering algorithms are able to find out \( p \) partitions of \( X \), \( C = \{C_1, C_2, C_3, \ldots, C_p\} \) such that

\[
C_j = \phi, \quad j = 1, 2, \ldots, p, \quad (1)
\]

\[
C_i \cap C_j = \phi, \quad i, j = 1, 2, \ldots, p \quad \text{and} \quad i \neq j, \quad (2)
\]

\[
\bigcup_{j=1}^{p} C_j = X. \quad (3)
\]
Generally, the similarity measures are used for assigning the data points to clusters. The commonly used similarity measure is Euclidean distance ($d_{ji}$) and is defined as

$$d_{ji} = \sqrt{\sum_{k=1}^{d} (x_{ik} - m_{jk})^2}, \quad (4)$$

where $x_i$ is a data point, $m_j$ is the cluster center of cluster $C_j$ and $d$ is the dimension space.

### 2.2. Multi-objective spotted hyena optimizer (MOSHO)

The multi-objective spotted hyena optimizer (MOSHO) is a multi-objective version of spotted hyena optimizer (SHO) proposed by Dhiman and Kumar in 2018. For multi-objectivity, two new elements in the existing SHO have been introduced. These are Archives and Group selection mechanism. These are described in preceding subsections.

#### 2.2.1. Archive

All the best-obtained Pareto optimal solutions are stored in a storage space known as an archive. It is evenly spread on Pareto front with concave, convex and disconnected fronts. It has two main components namely Controller and Grid.

##### 2.2.1.1. Archive controller

The inclusion of a particular solution in archive is decided by the controller. The updation rules for archive are as follows:

- The current solution is accepted if the archive is empty.
- The solution is automatically discarded if it is dominated by an individual within the archive.
- The solution is stored in the archive if none of the elements contained in the external population dominates such solution.
- The solutions are removed from the archive if these are dominated by the new element.

##### 2.2.1.2. Adaptive grid mechanism

The distributed Pareto fronts are obtained using adaptive grid mechanism. There are four linearly separable regions for defined objective function. The grid is responsible for the computation of each individual from the population if it lies outside the grid area. The grid space is formed as a result of uniform distribution of hypercubes.
2.2.1.3. Group selection mechanism

The major issue with multi-objective search space is to compare the new solutions with the existing solutions in search space. This issue is resorted using group selection mechanism. In group selection strategy, less crowded search space is populated with one of the best solutions in the group of nearby solutions using the roulette-wheel technique \( U_k \) defined as

\[
U_k = \frac{g}{N_k},
\]

where \( g \) is a constant variable with value greater than 1. \( N_k \) represents the number of Pareto optimal solutions to \( k \)th segment. This method popularly uses the classical method which defines the contribution of each individual using roulette wheel proportion.

3. Related Works

Bong and Rajeswari\(^{22}\) conducted a survey reporting the use of bio-inspired algorithms for solving clustering problems which increased many folds from 2006 to 2018. Lot of research has been done since the emergence of MOCK\(^{23}\)-based clustering algorithm which is used in various data partitioning problems. The advantage of MOCK is its high speed performance and ease of implementation. The drawback of this approach is the randomized initial centroids selection criteria.

Corne proposed Pareto front-based clustering algorithm for solving various real-life clustering problems and named it as PESA\(^{24}\) and PESA-II\(^{25}\). The main advantage of both these techniques is its multi-objectivity. The high computational cost is the main limitation of PESA and PESA-II methods. In continuation to the previous studies, Handl and Knowles proposed a new clustering algorithm which has new features such as scalability, automatic cluster identification and named it as Voronoi Initialized Evolutionary Nearest-Neighbor Algorithm (VIENNA)\(^{26}\).

Bandyopadhyay\(^{27}\) developed a new multi-objective clustering algorithm using NSGA-II\(^{28}\) and validated it for classifying remote sensing images. Similarly, MR brain images clustering application was developed using NSGA-II\(^{29}\). The fuzzy cluster validity indexes are simultaneously optimized to tackle the problem of fuzzy partitioning.

Santosh\(^{30}\) proposed MO ant-colonies-based clustering algorithm over the distributed environment which processed the data in parallel and meets the two objectives used for clustering. Immune-inspired MO algorithm was developed\(^{31}\) and further tested against UCI repository data set. In order to classify handwritten digits, MO clustering algorithm was proposed by Ma\(^{32}\) using immuno-dominance. Moreover, immune MO clustering algorithm was applied on SAR image partitioning in Ref. 33.

Saha and Bandyopadhyay\(^{34, 35}\) proposed a simulated annealing-based multi-objective clustering algorithm to find the optimal solution in a given data set which
Algorithm: Multi-objective Spotted Hyena Optimizer (MOSHO)

Input: Spotted hyenas population $S_h (h \leftarrow 1, 2, \ldots, n)$

Output: Archive of non-dominated optimal solutions

1: procedure MOSHO
2: Initialization of variables: $h, A, B,$ and $N$
3: Objective values for each search agent is computed
4: Initialize archive by finding non-dominated solutions
5: $O_h \leftarrow$ Exploration of best search agent from archive
6: $O_h \leftarrow$ group of found optimal solutions with respect to $A_h$ (archive)
7: while ($x < \text{Max}_{\text{iteration}}$) do
8: for each search agent do
9: Update the position of search agents
10: end for
11: Update $h, A, B,$ and $N$
12: Recalculate the fitness of all search agents
13: Again find the non-dominated solutions from updated pool of agents
14: Update archive with new solutions
15: if archive is full then
16: Grid method helps to exclude archive members that are responsible for overcrowding
17: New solutions are added to archive
18: end if
19: Adjust the search agents if they go beyond the search space
20: Calculate the fitness for each search agent
21: Update $A_h$ if better solutions are found.
22: Group updation is performed for $O_h$ with respect to archive
23: $x \leftarrow x + 1$
24: end while
25: return archive
26: end procedure

used symmetrical-distance. However, if the cost function is expensive to compute, the repeated annealing schedule is very slow.

Saha and Bandyopadhyay also proposed automatic multi-objective clustering technique (GenClustPESA2). The purpose of this method is to find automatic clusters and optimize together using a simulated annealing based technique (AMOSA). The computational complexity is the main drawback of this approach.

Abubaker et al. proposed a swarm intelligence-based clustering approach using multi-objective Particle Swarm Optimization and Simulated Annealing (MOPSOA). This algorithm showed better results than GenClustMOO and GenClusrPESA2. The main drawback of this approach is selecting large number of parameters during run time.
Xia\textsuperscript{38} proposed NSGA-II-based MO evolutionary technique named (MOEASSC), which can minimize WGS and negative information of weight entropy among clusters. An indicator is used known as projection similarity validity index (PSV Index) to discover the optimum number of clusters.

Suresh\textsuperscript{39} proposed a fuzzy clustering technique in a Multi-objective Optimization framework, which can automatically find the clusters in the data set. Hybrid MO algorithm (MOIMPSO) is proposed by Nanda and Panda\textsuperscript{40} which integrated the operators from particle swarm algorithm. Due to hybridization, the selection of parameters are large which can increase the computational efficiency.

Zhu\textsuperscript{41} proposed a multi-objective-based evolutionary soft subspace clustering technique (MOSSC). It can simultaneously optimize the weight between the cluster and weight within the cluster which can be separated within two different clustering validity criteria. The advantage of MOSSC is that it can combine the merits of soft-subspace clustering and properties of the multi-objective optimization-based approach for fuzzy clustering.

Manikandan and Selvarajan\textsuperscript{42} proposed a multi-objective clustering which is based on a hybrid optimization algorithm (MO-CS-PSO). This algorithm can be incorporated in the fitness function of the hybrid algorithm to enhance the efficiency and performance of the clustering problem.

Dasa\textsuperscript{43} developed an automatic kernel-based clustering technique with Multi-Elitist Particle Swarm Optimization Algorithm (MEPSO). The proposed algorithm is able to cluster the data sets without any prior knowledge of available groups in the data. This method is also based on the MEPSO model. A function which is known as kernel function was employed to make it possible for clustering on the data set.

Soto et al.\textsuperscript{44} proposed a wrapper feature selection algorithm using NSGA-II and SPEA2. To validate the performance of proposed technique, four different learning algorithms were trained. The local optima problem is the main drawback of this approach.

Zhu et al.\textsuperscript{45} proposed an improved feature selection algorithm approach (WFFSA) using the nature-inspired algorithm. Ranking of features was done using GA which prioritizes the features. However, WFFSA has limited performance when the number of features is increased.\textsuperscript{46}

Muni et al.\textsuperscript{47} proposed a feature selection using multi-tree GP algorithm (GPmtfs). For a given c-class problem, each classifier in GPmtfs is assigned c trees. The performance of GPmtfs is better than other methods using low computational efforts.

Wang et al.\textsuperscript{48} proposed a hybrid feature selection approach using PSO and rough sets theory. The particle is allocated as the relevance degree between the class labels and features.\textsuperscript{49} However, it overrules the use of rough sets in feature selection problems as they consume more computational time.

Esseghir et al.\textsuperscript{50} developed a filter-wrapper feature selection method using PSO which utilizes the strengths of both filters and wrappers approaches. Each particle is encoded using a score value in the proposed approach. Feature class dependency
is also ensured using filter approach. Performance of the proposed approach outperforms PSO-filter feature selection method.

Unler and Murat developed an adaptive selection-based wrapper feature selection algorithm. Feature importance or relevancy is not only articulated using likelihood Multi-Objective Evolutionary Algorithms but feature contribution in the overall classification is also considered as selection criteria.

Lin et al. developed a feature selection approach to optimize the parameters in vector machine and select the subset of features. The results state that the approach performed better than the competitive algorithm.

Liu et al. developed feature selection algorithm using multi-swarm PSO (MSPSO) algorithm to obtain the subset of features, moreover, parameter optimization is performed in SVM.

Sheng et al. proposed an adaptive niching method, which can dynamically adjust its parameter value depending on the problem instance as well as the search progress. It employed the three local searches of different features in a sophisticated manner to efficiently exploit the decision space. The experiments on both synthetic and real data show its superior performance.

Stoyanov et al. developed an approach in the field of chemoinformatics that had been applied to the problem of clustering of chemical databases. The proposed method has better discrimination power.

Zhu et al. proposed the multi-objective technique for feature selection in intrusion detection system (IDS). It uses two strategies: a special domination method and a predefined multiple targeted search. The selection of individuals are based on the bias-selection process which find the individuals along with limited number of selected features. This approach can alleviate the imbalance problem with higher classification accuracy.

Li et al. proposed kernel clustering algorithm for clustering problems. The kernel function can convert nonlinear data into high-dimensional feature space. In particular, three local learning operators are also designed to enhance the ability of exploration and exploitation.

Chen proposed a margin-based feature selection method for unsupervised learning and used the selected salient features to discover interesting clusters. It usually needs to partition data into clusters which are then used to select features in the feature selection process.

4. Proposed Approach

This section firstly discusses the main contribution of this work and proposed automatic clustering and feature selection approach.

4.1. Contribution

The main contribution of this paper is to automate the clustering and feature selection process using astrophysics concept. This paper presents a variable composite
agent representation scheme that helps to find clusters and feature subsets efficiently. The threshold setting for cluster centers and feature selection is proposed that inhibits the variance present in the given data set. The composite agent encodes cluster centers and features with their corresponding threshold values. A novel fitness function is proposed to compute the agent’s fitness values.

4.2. Astrophysics concept

The proposed technique is based on the fact that planets move in an elliptical path around the sun (see Fig. 1). The planets closer to the sun have the potential to become candidate solutions in exploitation. Far solutions can be used for exploration.
in search space. The elliptical shape helps in better exploration and exploitation. Here, we assume that the sun is considered as prey and planets are analogous to spotted hyenas. Figure 2(a) shows the spherical concept and Fig. 2(b) shows the elliptical orbit concepts which are responsible for both exploration and exploitation in MOSHO.

4.3. Astrophysics-based multi-objective spotted hyena optimizer (AMOSHO_CFS)

In this paper, the concept of paths followed by planets around the sun is adopted. Firstly, \( R \) numbers of spotted hyenas are randomly generated. Thereafter, the positions of these \( R \) spotted hyenas \((C_k, k = 1, 2, \ldots, R)\) are updated based on the position of best optimal solution \((C(x+1))\). Hence, the new positions of \( R \) spotted hyenas are computed as

\[
C(x+1) = \frac{O_h}{N} + (E_l(C_k) \times E_{\text{distance}}(0,1)),
\]

\[
E_l(C_k) = (C_1 \times C_2 \times \cdots \times C_R) \times \pi,
\]

where \( E_l(C_k) \) indicates the elliptical positions of spotted hyenas and \( E_{\text{distance}} \) is the distance between the positions of spotted hyenas to remove the conflict between each other.

4.4. Automatic clustering and feature selection using astrophysics based multi-objective spotted hyena optimizer (AMOSHO_CFS)

The proposed astrophysics based MOSHO for automatic clustering and feature selection (AMOSHO_CFS) approach consists of the following steps.

Step 1. The parameters such as number of population, maximum number of iterations, and number of clusters \((K_{\text{max}})\) are initialized.

Step 2. Search agents are initialized in such a way that they may contain the cluster centers with their corresponding threshold activation values (see Sec. 4.4.1).

Step 3. The active cluster centers and features are found by evaluating the threshold limits (see Sec. 4.4.4).

Step 4. Euclidean distance is calculated from all active cluster centers to each data point.

Step 5. Each data point is assigned to a cluster having minimum distance with respect to other clusters.

Step 6. However, if the data points are less than one (i.e. no data points belong to a particular cluster), then re-initialize the cluster centers of the agent using the procedure mentioned in Sec. 4.4.5.

Step 7. Fitness of agents is calculated using astrophysics-based MOSHO algorithm (see Sec. 4.4.6).
Step 8. The values of selected thresholds are assigned to clusters and features separately (see Sec. 4.4.2).

Step 9. The thresholds cutoffs are computed for each cluster center and feature based on selection threshold which is calculated in the previous step (see Sec. 4.4.3).

Step 10. Steps (3)–(9) are repeated until the termination criterion is satisfied.

Step 11. The best search agent will return the optimal feature subset, cluster centers, and number of clusters.

4.4.1. Agent representation and initialization

The proposed AMOSHO.CFS employed a variable composite agent representation approach to encode both features and cluster centers. An agent consists of real numbers of dimension \((K_{\text{max}} + D) + (K_{\text{max}} \times D)\) for \(N\) number of data points, \(D\) dimensions, and user-defined \(K_{\text{max}}\) clusters. The \(K_{\text{max}}\) is set as follows:

\[
K_{\text{max}} = \sqrt{D}.
\]  

(8)

The values assigned to \((K_{\text{max}} + D)\) are positive numbers in range \([0, 1]\). The first \(K_{\text{max}}\) entry indicates that the belonging cluster is activated or not during clustering process. \(D\) entry indicates whether the features are activated or not. \((K_{\text{max}} \times D)\) represents \(K_{\text{max}}\) cluster centers of \(D\) dimension. The vector \(V_i(t)\) of \(i^{th}\) agent at particular time \(t\) is shown in Fig. 3 where \(c_{rij}\) is the \(j^{th}\) cluster center of \(i^{th}\) agent. \(THC_{ij}\) represents the value of threshold of cluster center \(c_{rij}\). \(THF_{ij}\) is the value of threshold of \(j^{th}\) feature of \(i^{th}\) agent. \(THC\) and \(THF\) are selection thresholds for selecting active cluster centers and features, respectively.

Example 1. Let us assume that the number of calculated clusters \(K_{\text{max}} = 4\) in three-dimensional space, i.e. \(D = 3\).

The first entries (i.e. 0.5, 0.6, 0.3 and 0.2) indicate the cluster selection thresholds with respect to four clusters. Then, the next three entries (i.e. 0.6, 0.2 and 0.7) represent the feature selection thresholds and the last entries represent four cluster centers, i.e. (4.8, 3.1 and 1.5), (5.6, 4.3 and 0.9), (6.8, 2.9 and 4.8) and (7.6, 3.0, 2.3).

![Fig. 3. Agent representation scheme.](media/1850385-10)
4.4.2. Threshold setting computation

A novel threshold setting method for cluster center computation is proposed which is based on within-cluster variation and is defined as

\[ THC_l = \left( \frac{1}{n_l} \sum_{j=1}^{n_l} (x_j^l - C_l)^2 \right)^{\frac{1}{2}}, \]

where \( l = 2, 3, \ldots, K_{\text{max}} \). \( THC_l \) is the selection threshold with respect to cluster \( l \). \( n_l \) represents the number of data points and \( x_j^l \) represents the \( j \)th data point that belongs to cluster \( C_l \).

To determine the number of clusters automatically, it is necessary that the same feature corresponds to each cluster. Hence, the threshold setting mechanism is proposed for finding an efficient number of features. It is computed as

\[ THF_p = \frac{1}{K} \sum_{j=1}^{K} \left( \frac{V_{r_p} - V_{r_p,j}}{V_{r_p}} \right), \quad \text{where } p = 1, 2, \ldots, D, \]

where \( THF_p \) is threshold value of \( p \)th feature and \( V_{r_p} \) is variance of \( p \)th feature. \( V_{r_p,j} \) represents the variance of \( p \)th feature in \( j \)th cluster. \( K \) represents the clusters that are used for partitioning the data set. \( THF_p \) is the average value of significance of \( p \)th feature.

4.4.3. Threshold cutoff computation

The cutoff feature and cluster threshold values are set according to their mean values. These are computed over all dimensions and clusters in case of feature thresholds and cluster thresholds, respectively. The cutoff threshold for cluster selection is defined as

\[ T_{\text{cutt,clus}} = \frac{1}{K} \sum_{l=1}^{K} THC_l. \]

The cutoff threshold for feature selection is defined as

\[ T_{\text{cutt,feat}} = \frac{1}{D} \sum_{p=1}^{D} THF_p. \]

In the initial stage of algorithm, the cutoff thresholds \( T_{\text{cutt,clus}} \) and \( T_{\text{cutt,feat}} \) are set to 0.5.

**Example 2.** The cluster center and feature selection thresholds are considered as given in Example 1 (i.e. 0.5, 0.6, 0.3, 0.2) and (0.6, 0.2, 0.7), respectively. Based on the threshold cutoff computation approach, \( T_{\text{cutt,clus}} \) and \( T_{\text{cutt,feat}} \) are 0.5, respectively.
4.4.4. Active cluster center and feature extraction

The selection of cluster centers and features in an agent depend upon their cutoff threshold values, i.e. $T_{\text{cutt,clus}}$ and $T_{\text{cutt,feat}}$. The rules for active cluster center and feature extraction are described as

\[ \text{If } THC_{i,j} > T_{\text{cutt,clus}} \text{ then} \]
\[ \text{ } j \text{th cluster in the } i \text{th agent is active (i.e. } m_{i,j}) \]
\[ \text{Else} \]
\[ \text{ } j \text{th cluster in the } i \text{th agent is not active} \]
\[ \text{EndIf} \]

\[ \text{If } THF_{i,j} > T_{\text{cutt,feat}} \text{ then} \]
\[ \text{ } j \text{th feature in cluster centers for } i \text{th agent is active} \]
\[ \text{Else} \]
\[ \text{ } j \text{th feature in cluster centers for } i \text{th agent is not active} \]
\[ \text{EndIf} \]

There is a possibility that no thresholds are greater than the cutoff threshold values, i.e. $T_{\text{cutt,clus}}$ and $T_{\text{cutt,feat}}$. In this situation, two threshold values are selected randomly and re-initialized.

**Example 3.** The cluster center and feature selection thresholds are $(0.5, 0.6, 0.3, 0.2)$ and $(0.6, 0.2, 0.7)$, respectively. The thresholds cutoff for both cluster center and feature selection is 0.5. Firstly, the cluster center thresholds are employed to select the number of clusters. Only two threshold values (i.e. 0.5 and 0.6) are greater than 0.5. Therefore, the thresholds of feature selection are used to choose the features from their corresponding clusters (see Fig. 4). Hence, two thresholds (i.e. 0.6 and 0.7) are greater than the cutoff threshold value (i.e. 0.5) for feature selection.

![Proposed agent encoding scheme](image)

**Fig. 4.** Proposed agent encoding scheme.

4.4.5. Cluster center validation

The proposed technique may suffer from empty clusters problems. It means that the cluster has no data point. This issue generally arises because of misplacement of cluster center as compared to data distribution. To deal with empty clusters, the
re-initialization of cluster centers can be done. The average computation formula is used to allocate \( n/K \) data points to the nearest cluster center.

**Example 4.** Let there be 150 data points having 3-dimensional feature space in a given data set. Assume there are three active number of clusters and cluster centers for a particular agent \(((0.9, 0.42, -0.01), (4.0, 3.1, 0.8), (6.1, 1.2, 0.8))\). Here, cluster center \((0.9, 0.42, -0.01)\) have no data point assignment. This is the case with outside cluster data point allocation. Average computational formulae are used to assign data points to nearest clusters. The updated cluster centers are \((3.3128, 2.269, 0.601), (2.900, 1.832, 0.266)\) and \((4.789, 2.456, 0.876)\).

4.4.6. **Fitness function evaluation**

In this paper, we have chosen \(I\)-index\(^{20}\) and Sym-index\(^{61}\) for clustering criteria. The first quantifies the amount of symmetry present in a particular partitioning. The latter measures the compactness of partitioning. These are mathematically described as

\[
I(K) = \left( \frac{1}{K} \times \frac{ED_1}{ED_K} \times D_K \right)^q.
\]

Here,

\[
ED_K = \sum_{i=1}^{K} \sum_{j=1}^{n_K} Dst_e(x^K_j, m_i), \quad D_K = \max_{i,j=1} Dst_e(m_i, m_j),
\]

(13)

where \(K\) is the number of clusters for partitioning the data set. The value of \(q\) is set to 2. \(D_K\) is the maximum Euclidean distance between two center clusters. \(Dst_e\) represents the difference between \(m_i\) and \(m_j\). \(Dst_e(x^K_j, m_i)\) is calculated as

\[
Dst_e(x^K_j, m_i) = Dist_{sym}(x, p_i) \times Dist_{e}(x, m_i).
\]

(14)

where \(m_i\) is the cluster center of particular cluster \(i\). \(Dist_{e}(x, m_i)\) is the Euclidean distance between point \(x\) and \(m_i\). \(I\)-index value is responsible for better clustering solutions.

\[
Sym(K) = \left( \frac{1}{K} \times \frac{1}{ED_K} \times D_K \right), \quad ED_K = \sum_{i=1}^{K} E_i,
\]

\[E_i = \sum_{j=1}^{n_K} D^*_p(\bar{x}^K_j, \bar{m}_i), \quad D_K = \max_{i,j=1} Dst_e(m_i, m_j),\]

(15)

The penalty function is introduced to find the optimal number of clusters during run time, and mathematically defined as

\[
P_{term1} = K_{max} - K.
\]

(16)

Most of the clustering criteria do not consider the appropriate number of features. Hence, another penalty function is employed for selecting the optimal number of
features $d$ from total features $D$. It is defined as

$$P_{\text{term} \ 2} = \frac{D - d}{D - 1}. \tag{17}$$

By combining these terms, the new clustering criterion is defined as

$$\text{Fit} = I(K) \times \text{Sym}(K) \times P_{\text{term} \ 1} \times P_{\text{term} \ 2}. \tag{18}$$

4.5. **Computational complexity**

The space and time complexities of AMOSHO_CFS are discussed below.

4.5.1. **Time complexity**

1. Initialization of AMOSHO_CFS population needs $O(a_s \times s_l)$ time where $a_s$ is the number of agents and $s_l$ is the distance of each agent which is encoded, respectively.
2. It requires $O(a_s \times K_{\text{max}})$ for active cluster and feature extraction.
3. The data points assigned to different clusters require $O(n^2 \times K_{\text{max}})$.
4. For fitness calculation, the algorithm requires $O(n \times K_{\text{max}})$ time.
5. Steps 3 and 4 are repeated for all agents until the termination criteria is satisfied.

Hence, the total time complexity of AMOSHO_CFS algorithm for maximum number of iterations is $O(n^2 \times K_{\text{max}} \times a_s \times \text{Max}_{\text{iteration}})$.

4.5.2. **Space complexity**

The space complexity of the proposed algorithm is computed during its initial step. Therefore, the space complexity of the proposed AMOSHO_CFS algorithm is $O(a_s \times s_l)$.

5. **Experimental Results and Discussions**

This section presents the performance of proposed approach on eight real-life benchmark functions. The results are evaluated and compared with four well-known multi-objective algorithms namely FeaClusMOO,$^{19}$ FeaClusMOO_EUC,$^{19}$ VGAPS$^{20}$ and K-means.$^{21}$

5.1. **Data sets used**

The efficiency and effectiveness of proposed AMOSHO_CFS is tested on eight real-life data sets namely Iris, Wine, Glass, Breast Cancer, Vowel, Haberman, Bupa and CMC. The characteristics of these data sets are tabulated in Table 1.$^{62}$
Table 1. Description of data sets.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>No. of data points</th>
<th>No. of features</th>
<th>No. of classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>150</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Wine</td>
<td>178</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>Glass</td>
<td>214</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Breast cancer</td>
<td>683</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>Vowel</td>
<td>871</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>Haberman</td>
<td>306</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Bupa</td>
<td>345</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>CMC</td>
<td>1473</td>
<td>9</td>
<td>3</td>
</tr>
</tbody>
</table>

5.2. Experimental setup

The algorithms are implemented and tested in Matlab R2016a version. The platform used for experimentation is Microsoft Windows 10 having 64 bits and i7 core processor. The search agents and maximum number of iterations of the proposed approach are set as 30 and 100, respectively.

5.3. Relevance of selected features

The selected features are estimated by the frequency of selected features and their Laplacian score (LS). The feature frequency is defined as

\[ f_r(N_{f_i}) = \frac{R_{N_{f_i}}}{R_I}, \]  

(19)

where \( R_{N_{f_i}} \) represents the number of times a particular feature is selected and \( R_I \) is the total number of independent runs. Table 2 shows the features frequency selected from AMOSHO_CFS and their LS for eight data sets. From Table 2, it is shown that AMOSHO_CFS selects features 3 and 4 of Iris data set. These features have high LS value. For Wine data set, AMOSHO_CFS selects features 1, 6, 7 and 13. Similarly, for Glass data set, the frequently selected features are 1, 3, 4 and 7. For Haberman data set, the selected features are 1 and 3. For Bupa data set, AMOSHO_CFS selects features 2 and 5. For Cancer data set, the selected features are 1, 2, 3, 6 and 7. For Vowel data set, the selected feature is 2. For CMC data set, the features 1 and 4 are selected.

5.4. Relevance of selected clusters

The frequency of number of clusters with their classification accuracy (AC) values are selected to differentiate the important clusters from others. The mathematical formulation of frequency of clusters is as follows:

\[ F_r(N_{c_i}) = \frac{R_{N_{c_i}}}{R_I}, \]  

(20)

where \( R_{N_{c_i}} \) represents the number of times a particular cluster is selected and \( R_I \) is the total number of independent runs. Table 3 demonstrates the count of selected
Table 2. The obtained features frequency and their corresponding Laplacian score (LS) using AMOSHO_CFS algorithm.

<table>
<thead>
<tr>
<th>Feature number</th>
<th>Data sets 1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>fr</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>LS</td>
<td>0.51</td>
<td>0.84</td>
<td>0.98</td>
<td>0.96</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Wine</td>
<td>fr</td>
<td>0.83</td>
<td>0.24</td>
<td>0.05</td>
<td>0.30</td>
<td>0.18</td>
<td>0.74</td>
<td>0.05</td>
<td>0.05</td>
<td>0.30</td>
<td>0.18</td>
<td>0.18</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>LS</td>
<td>0.64</td>
<td>0.14</td>
<td>0.10</td>
<td>0.23</td>
<td>0.54</td>
<td>0.38</td>
<td>0.45</td>
<td>0.13</td>
<td>0.15</td>
<td>0.18</td>
<td>0.17</td>
<td>0.30</td>
</tr>
<tr>
<td>Glass</td>
<td>fr</td>
<td>0.92</td>
<td>0.00</td>
<td>1.00</td>
<td>0.93</td>
<td>0.00</td>
<td>0.05</td>
<td>0.81</td>
<td>0.17</td>
<td>0.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>LS</td>
<td>0.80</td>
<td>0.75</td>
<td>0.94</td>
<td>0.71</td>
<td>0.70</td>
<td>0.50</td>
<td>0.91</td>
<td>0.55</td>
<td>0.14</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Haberman</td>
<td>fr</td>
<td>0.22</td>
<td>0.00</td>
<td>0.10</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>LS</td>
<td>0.91</td>
<td>0.88</td>
<td>0.95</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Bupa</td>
<td>fr</td>
<td>0.13</td>
<td>0.80</td>
<td>0.46</td>
<td>0.04</td>
<td>0.70</td>
<td>0.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>LS</td>
<td>0.70</td>
<td>0.88</td>
<td>0.82</td>
<td>0.70</td>
<td>0.95</td>
<td>0.27</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Cancer</td>
<td>fr</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.68</td>
<td>0.00</td>
<td>0.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>LS</td>
<td>0.91</td>
<td>0.93</td>
<td>0.92</td>
<td>0.90</td>
<td>0.83</td>
<td>0.95</td>
<td>0.88</td>
<td>0.92</td>
<td>0.76</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Vowel</td>
<td>fr</td>
<td>0.10</td>
<td>1.00</td>
<td>0.08</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>LS</td>
<td>0.90</td>
<td>0.98</td>
<td>0.97</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>CMC</td>
<td>fr</td>
<td>0.89</td>
<td>0.08</td>
<td>0.08</td>
<td>0.80</td>
<td>0.08</td>
<td>0.00</td>
<td>0.13</td>
<td>0.00</td>
<td>0.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>LS</td>
<td>0.96</td>
<td>0.76</td>
<td>0.70</td>
<td>0.90</td>
<td>0.35</td>
<td>0.30</td>
<td>0.70</td>
<td>0.70</td>
<td>0.22</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Table 3. The obtained clusters frequency and their corresponding classification accuracy (AC) using AMOSHO_CFS algorithm.

<table>
<thead>
<tr>
<th>Number of clusters</th>
<th>Data sets 2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>Fr</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>AC</td>
<td>0.95</td>
<td>0.80</td>
<td>0.68</td>
<td>0.60</td>
<td>0.51</td>
<td>0.50</td>
<td>0.45</td>
<td>0.45</td>
<td>0.40</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Wine</td>
<td>Fr</td>
<td>0.85</td>
<td>0.08</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>AC</td>
<td>0.65</td>
<td>0.55</td>
<td>0.53</td>
<td>0.37</td>
<td>0.37</td>
<td>0.32</td>
<td>0.30</td>
<td>0.30</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Glass</td>
<td>Fr</td>
<td>0.00</td>
<td>0.00</td>
<td>0.17</td>
<td>0.65</td>
<td>0.12</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>AC</td>
<td>0.36</td>
<td>0.40</td>
<td>0.40</td>
<td>0.80</td>
<td>0.55</td>
<td>0.52</td>
<td>0.52</td>
<td>0.48</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Haberman</td>
<td>Fr</td>
<td>0.26</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>AC</td>
<td>0.45</td>
<td>0.35</td>
<td>0.26</td>
<td>0.26</td>
<td>0.26</td>
<td>0.22</td>
<td>0.20</td>
<td>0.17</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Bupa</td>
<td>Fr</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>AC</td>
<td>0.45</td>
<td>0.43</td>
<td>0.37</td>
<td>0.34</td>
<td>0.33</td>
<td>0.30</td>
<td>0.27</td>
<td>0.20</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Cancer</td>
<td>Fr</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>AC</td>
<td>0.95</td>
<td>0.75</td>
<td>0.70</td>
<td>0.65</td>
<td>0.62</td>
<td>0.55</td>
<td>0.50</td>
<td>0.50</td>
<td>0.45</td>
<td>0.42</td>
<td>0.30</td>
</tr>
<tr>
<td>Vowel</td>
<td>Fr</td>
<td>0.00</td>
<td>0.00</td>
<td>0.09</td>
<td>0.80</td>
<td>0.08</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>AC</td>
<td>0.30</td>
<td>0.36</td>
<td>0.45</td>
<td>0.55</td>
<td>0.50</td>
<td>0.45</td>
<td>0.42</td>
<td>0.40</td>
<td>0.35</td>
<td>0.34</td>
<td>0.35</td>
</tr>
<tr>
<td>CMC</td>
<td>Fr</td>
<td>0.90</td>
<td>0.04</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>AC</td>
<td>0.40</td>
<td>0.30</td>
<td>0.25</td>
<td>0.23</td>
<td>0.20</td>
<td>0.17</td>
<td>0.15</td>
<td>0.15</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

clusters and their corresponding classification accuracy for eight data sets such as Iris, Wine, Glass, Haberman, Bupa, Cancer, Vowel, and CMC. For Iris data set, the proposed AMOSHO_CFS generates three clusters in each run with its corresponding high classification accuracy. For Wine data set, AMOSHO_CFS generates three
Astrophysics inspired multi-objective approach for automatic clustering

clusters based on classification accuracy. The cluster count for Glass data set is 5. The cluster count for Haberman data set is 2. The number of clusters for Bupa data set is 2 and their cluster accuracy is high. For Cancer data set, proposed AMOSHO_CFS produces 6 clusters. For Vowel data set, the number of clusters is 2. The cluster count for CMC data set is 3 with high classification accuracy.

Table 4. The obtained mean and standard deviation of cluster quality measures using the proposed and other competitive algorithms.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Methods</th>
<th>No. of clusters</th>
<th>No. of features</th>
<th>Classification accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>AMOSHO_CFS</td>
<td>3.0 (0.0)</td>
<td>2.0 (0.0)</td>
<td>96.5 (0.4)</td>
</tr>
<tr>
<td></td>
<td>KM</td>
<td>3.0 (0.0)</td>
<td>2.7 (0.9)</td>
<td>94.8 (2.4)</td>
</tr>
<tr>
<td></td>
<td>VGAPS</td>
<td>3.0 (0.0)</td>
<td>2.2 (0.3)</td>
<td>95.4 (1.9)</td>
</tr>
<tr>
<td></td>
<td>FeaClusMOO_EUC</td>
<td>3.0 (0.0)</td>
<td>2.9 (0.2)</td>
<td>95.9 (1.5)</td>
</tr>
<tr>
<td></td>
<td>FeaClusMOO</td>
<td>3.0 (0.0)</td>
<td>2.0 (0.3)</td>
<td>96.0 (1.6)</td>
</tr>
<tr>
<td>Wine</td>
<td>AMOSHO_CFS</td>
<td>3.05 (0.6)</td>
<td>3.8 (1.1)</td>
<td>66.4 (4.1)</td>
</tr>
<tr>
<td></td>
<td>KM</td>
<td>3.8 (2.3)</td>
<td>6.2 (2.6)</td>
<td>64.7 (9.1)</td>
</tr>
<tr>
<td></td>
<td>VGAPS</td>
<td>3.2 (2.0)</td>
<td>5.7 (2.9)</td>
<td>65.0 (8.7)</td>
</tr>
<tr>
<td></td>
<td>FeaClusMOO_EUC</td>
<td>3.5 (1.1)</td>
<td>5.8 (1.5)</td>
<td>65.6 (8.9)</td>
</tr>
<tr>
<td></td>
<td>FeaClusMOO</td>
<td>3.4 (2.5)</td>
<td>6.0 (3.1)</td>
<td>65.2 (9.8)</td>
</tr>
<tr>
<td>Glass</td>
<td>AMOSHO_CFS</td>
<td>5.9 (0.8)</td>
<td>3.8 (0.8)</td>
<td>52.2 (3.2)</td>
</tr>
<tr>
<td></td>
<td>KM</td>
<td>6.2 (1.9)</td>
<td>4.3 (1.6)</td>
<td>44.3 (4.7)</td>
</tr>
<tr>
<td></td>
<td>VGAPS</td>
<td>6.0 (1.2)</td>
<td>4.0 (0.6)</td>
<td>44.7 (5.1)</td>
</tr>
<tr>
<td></td>
<td>FeaClusMOO_EUC</td>
<td>5.9 (1.5)</td>
<td>3.5 (1.1)</td>
<td>43.1 (5.4)</td>
</tr>
<tr>
<td></td>
<td>FeaClusMOO</td>
<td>5.7 (2.0)</td>
<td>3.0 (1.9)</td>
<td>46.4 (4.9)</td>
</tr>
<tr>
<td>Haberman</td>
<td>AMOSHO_CFS</td>
<td>2.3 (0.4)</td>
<td>2.0 (0.0)</td>
<td>62.5 (9.7)</td>
</tr>
<tr>
<td></td>
<td>KM</td>
<td>2.9 (1.7)</td>
<td>2.0 (0.0)</td>
<td>56.7 (6.4)</td>
</tr>
<tr>
<td></td>
<td>VGAPS</td>
<td>2.2 (0.6)</td>
<td>2.9 (0.6)</td>
<td>56.3 (5.8)</td>
</tr>
<tr>
<td></td>
<td>FeaClusMOO_EUC</td>
<td>2.5 (0.5)</td>
<td>2.3 (0.5)</td>
<td>55.8 (8.1)</td>
</tr>
<tr>
<td></td>
<td>FeaClusMOO</td>
<td>2.7 (1.2)</td>
<td>2.7 (0.9)</td>
<td>56.9 (7.5)</td>
</tr>
<tr>
<td>Bupa</td>
<td>AMOSHO_CFS</td>
<td>2.2 (0.3)</td>
<td>2.2 (0.4)</td>
<td>54.7 (2.2)</td>
</tr>
<tr>
<td></td>
<td>KM</td>
<td>2.9 (1.2)</td>
<td>2.5 (0.4)</td>
<td>52.4 (2.4)</td>
</tr>
<tr>
<td></td>
<td>VGAPS</td>
<td>2.6 (1.1)</td>
<td>3.0 (0.5)</td>
<td>51.7 (3.1)</td>
</tr>
<tr>
<td></td>
<td>FeaClusMOO_EUC</td>
<td>3.0 (0.7)</td>
<td>2.4 (0.8)</td>
<td>51.5 (3.9)</td>
</tr>
<tr>
<td></td>
<td>FeaClusMOO</td>
<td>2.6 (1.9)</td>
<td>2.9 (1.2)</td>
<td>52.1 (2.5)</td>
</tr>
<tr>
<td>Cancer</td>
<td>AMOSHO_CFS</td>
<td>2.0 (0.0)</td>
<td>3.8 (0.9)</td>
<td>95.2 (0.3)</td>
</tr>
<tr>
<td></td>
<td>KM</td>
<td>2.4 (1.6)</td>
<td>4.3 (1.5)</td>
<td>94.3 (2.1)</td>
</tr>
<tr>
<td></td>
<td>VGAPS</td>
<td>2.4 (1.3)</td>
<td>3.9 (2.0)</td>
<td>93.9 (2.6)</td>
</tr>
<tr>
<td></td>
<td>FeaClusMOO_EUC</td>
<td>2.2 (0.4)</td>
<td>4.1 (1.4)</td>
<td>94.6 (2.9)</td>
</tr>
<tr>
<td></td>
<td>FeaClusMOO</td>
<td>2.0 (1.1)</td>
<td>4.0 (1.8)</td>
<td>94.8 (1.2)</td>
</tr>
<tr>
<td>Vowel</td>
<td>AMOSHO_CFS</td>
<td>6.0 (0.6)</td>
<td>2.1 (0.3)</td>
<td>55.5 (2.2)</td>
</tr>
<tr>
<td></td>
<td>KM</td>
<td>6.2 (0.9)</td>
<td>2.0 (0.0)</td>
<td>53.9 (4.5)</td>
</tr>
<tr>
<td></td>
<td>VGAPS</td>
<td>6.3 (1.6)</td>
<td>2.9 (0.7)</td>
<td>52.8 (3.9)</td>
</tr>
<tr>
<td></td>
<td>FeaClusMOO_EUC</td>
<td>6.0 (1.4)</td>
<td>2.1 (0.3)</td>
<td>53.4 (3.6)</td>
</tr>
<tr>
<td></td>
<td>FeaClusMOO</td>
<td>6.0 (2.0)</td>
<td>2.8 (0.6)</td>
<td>54.1 (4.1)</td>
</tr>
<tr>
<td>CMC</td>
<td>AMOSHO_CFS</td>
<td>3.0 (0.5)</td>
<td>2.2 (0.4)</td>
<td>43.0 (3.1)</td>
</tr>
<tr>
<td></td>
<td>KM</td>
<td>3.5 (1.1)</td>
<td>3.9 (1.2)</td>
<td>41.8 (3.8)</td>
</tr>
<tr>
<td></td>
<td>VGAPS</td>
<td>3.2 (1.9)</td>
<td>3.8 (1.5)</td>
<td>41.5 (4.3)</td>
</tr>
<tr>
<td></td>
<td>FeaClusMOO_EUC</td>
<td>2.9 (0.8)</td>
<td>3.6 (1.4)</td>
<td>40.4 (5.1)</td>
</tr>
<tr>
<td></td>
<td>FeaClusMOO</td>
<td>3.2 (1.4)</td>
<td>3.0 (1.0)</td>
<td>41.9 (4.2)</td>
</tr>
</tbody>
</table>
5.5. Performance evaluation

Table 4 shows the mean and standard deviation of cluster quality measures obtained from the proposed and other clustering algorithms.

For Iris data set, AMOSHO_CFS obtains the optimal accuracy of classification as compared to other competitor approaches such as FeaClusMOO, FeaClusMOO_EUC, VGAPS, and KM. The proposed algorithm selects the features 3 and 4 during simulation run. For Wine data set, AMOSHO_CFS achieves the best classification accuracy as compared to other competitive methods. The number of clusters and number of features obtained from AMOSHO_CFS are 3 and 4, respectively.

For Glass data set, the proposed AMOSHO_CFS algorithm produces the number of clusters as 6. The number of features obtained from FeaClusMOO, FeaClusMOO_EUC, VGAPS, KM, and AMOSHO_CFS is 4. For Haberman data set, AMOSHO_CFS provides the optimal classification accuracy. The classification accuracies obtained from FeaClusMOO and FeaClusMOO_EUC are 56.9% and 55.8%, respectively. The classification accuracies obtained from KM and VGAPS are 56.7% and 56.3%, respectively. For Bupa data set, AMOSHO_CFS obtains the number of clusters as 2 and number of features as 2. Therefore, the classification accuracy obtained from proposed AMOSHO_CFS is better than the other approaches. For Cancer data set, all the above-mentioned approaches produce the accurate number of clusters and features. The classification accuracy obtained from AMOSHO_CFS is better than other algorithms. For Vowel and CMC data sets, AMOSHO_CFS produces best classification accuracy followed by FeaClusMOO, FeaClusMOO_EUC, VGAPS, and KM.

5.6. Statistical testing

The comparison of algorithms does not guarantee the efficacy and superiority of the proposed algorithm. The possibility of getting good results, by chance, cannot be ignored. For this, Wilcoxon’s rank sum test is performed at 5% level of significance and the p-values are tabulated in Table 5. The results show that the classification accuracy obtained from the proposed AMOSHO_CFS is statistically significant as compared to other algorithms.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>KM</th>
<th>VGAPS</th>
<th>FeaClusMOO_EUC</th>
<th>FeaClusMOO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>3.54E-05</td>
<td>1.03E-04</td>
<td>2.70E-02</td>
<td>2.43E-02</td>
</tr>
<tr>
<td>Wine</td>
<td>6.04E-01</td>
<td>5.71E-01</td>
<td>2.43E-01</td>
<td>3.17E-01</td>
</tr>
<tr>
<td>Glass</td>
<td>1.57E-01</td>
<td>1.20E-03</td>
<td>4.71E-04</td>
<td>1.42E-05</td>
</tr>
<tr>
<td>Haberman</td>
<td>2.51E-04</td>
<td>4.80E-04</td>
<td>7.20E-02</td>
<td>2.80E-03</td>
</tr>
<tr>
<td>Bupa</td>
<td>1.20E-04</td>
<td>8.42E-02</td>
<td>1.23E-04</td>
<td>1.67E-05</td>
</tr>
<tr>
<td>Cancer</td>
<td>1.18E-04</td>
<td>1.13E-05</td>
<td>1.35E-03</td>
<td>7.50E-02</td>
</tr>
<tr>
<td>Vowel</td>
<td>1.45E-03</td>
<td>8.77E-02</td>
<td>6.44E-02</td>
<td>1.60E-03</td>
</tr>
<tr>
<td>CMC</td>
<td>1.80E-04</td>
<td>1.19E-03</td>
<td>2.20E-03</td>
<td>1.30E-02</td>
</tr>
</tbody>
</table>
6. AMOSHO_CFS for Real-Life Applications

The proposed AMOSHO_CFS is validated and tested on two real-life problems such as image segmentation and microarray data analysis.

6.1. Image segmentation

Image segmentation is a technique of decomposing an image into different regions with similar features. Image segmentation can be formulated as a clustering problem. It is mathematically defined as follows:

\[
\sum_{i=1}^{n} R_i = I
\]  

such that

\[
R_i \cap R_j = \phi. 
\]

The proposed algorithm is tested on four color images, namely, Mandrill, Airplane (F-16), Peppers and House. The size of each image is 512 × 512. The number of data points for each image are 262144. The original images and clustered images are shown in Fig. 5. For afore-mentioned images, the optimal range of clusters is given in Ref. 6. Table 6 shows the results obtained from the proposed and other clustering techniques on color images. For Mandrill and Airplane (F-16) images, AMOSHO_CFS generate six clusters. AMOSHO_CFS gives seven and five partitions in case of Peppers and House images, respectively. The results are compared with
Table 6. Average and standard deviation of clustering results obtained from different clustering algorithms on four color images over 40 independent runs.

<table>
<thead>
<tr>
<th>Image</th>
<th>Cluster range</th>
<th>AMOSHO_CFS</th>
<th>KM</th>
<th>VGAPS</th>
<th>FeaClusMOO_EUC</th>
<th>FeaClusMOO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandrill</td>
<td>5–10</td>
<td>6.000(0.696)</td>
<td>4.100(0.201)</td>
<td>4.001(0.126)</td>
<td>4.524(0.400)</td>
<td>3.000(0.000)</td>
</tr>
<tr>
<td>Airplane</td>
<td>5–10</td>
<td>6.010(0.000)</td>
<td>4.017(0.750)</td>
<td>3.200(0.075)</td>
<td>6.310(0.090)</td>
<td>3.498(0.055)</td>
</tr>
<tr>
<td>(F-16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Peppers</td>
<td>5–10</td>
<td>7.007(0.000)</td>
<td>6.007(0.028)</td>
<td>6.810(0.060)</td>
<td>3.152(0.366)</td>
<td>5.007(0.060)</td>
</tr>
<tr>
<td>House</td>
<td>5–10</td>
<td>5.017(1.203)</td>
<td>5.285(0.070)</td>
<td>4.657(0.674)</td>
<td>7.400(0.041)</td>
<td>5.225(0.013)</td>
</tr>
</tbody>
</table>

Table 7. Unpaired $t$-test between the best outperforming and second best outperforming algorithms for color images.

<table>
<thead>
<tr>
<th>Image</th>
<th>Standard error</th>
<th>$t$</th>
<th>95% confidence interval</th>
<th>Two-tailed $P$</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandrill</td>
<td>0.095</td>
<td>2.5800</td>
<td>$-0.44586$ to $-0.05211$</td>
<td>0.0121</td>
<td>Significant</td>
</tr>
<tr>
<td>Airplane</td>
<td>0.196</td>
<td>5.4102</td>
<td>$-1.51003$ to $-0.67569$</td>
<td>$&lt;0.0001$</td>
<td>Significant</td>
</tr>
<tr>
<td>Peppers</td>
<td>0.078</td>
<td>2.5204</td>
<td>$-0.44125$ to $-0.05115$</td>
<td>0.0112</td>
<td>Significant</td>
</tr>
<tr>
<td>House</td>
<td>0.230</td>
<td>1.7526</td>
<td>$-0.86160$ to 0.06160</td>
<td>0.0864</td>
<td>Not significant</td>
</tr>
</tbody>
</table>

KM, VGAPS, FeaClusMOO_EUC, and FeaClusMOO algorithms. Table 7 shows the significance of AMOSHO_CFS using unpaired $t$-test. These results show the supremacy of proposed AMOSHO_CFS over other approaches.

6.2. Microarray data analysis

The proposed technique is also tested on microarray data sets for analysis. The three well-known data sets are employed for experimentation. These data sets are Yeast Sporulation, Human Fibroblasts Serum, and Rat CNS. The results are tested and compared with AMOSHO_CFS, KM, VGAPS, FeaClusMOO_EUC and FeaClusMOO algorithms. The parameter settings of above-mentioned algorithms are set same as in their original papers.

Silhouette index (SC) is used to measure the separation and compactness of clusters. The higher value of SC means the better performance of clustering algorithm. However, all the clustering algorithms have been run 10 times independently.

Figure 6 shows the number of clusters and average values of Silhouette index obtained by the proposed and competitive algorithms. The results reveal that AMOSHO_CFS has generated 5 clusters for Yeast Sporulation data set. KM, VGAPS and FeaClusMOO_EUC yielded 6 clusters. The number of clusters generated by FeaClusMOO is seven. The proposed AMOSHO_CFS produces the better value of SC as 0.6985.

For Human Fibroblasts Serum data set, the proposed AMOSHO_CFS and FeaClusMOO_EUC produced the number of clusters as 4. KM and FeaClusMOO generated eight clusters. VGAPS produced the number of clusters as 6. The value of SC obtained from AMOSHO_CFS is 0.4003.
Astrophysics inspired multi-objective approach for automatic clustering

Fig. 6. (Color online) Comparison of proposed AMOSHO_CFS approach with competitor approaches in terms of: (a) number of clusters; (b) SC values.

For Rat CNS data set, the proposed AMOSHO_CFS generated the number of clusters as 3. KM and VGAPS produced the number of clusters as 5. The number of clusters obtained from both FeaClusMOO_EUC and FeaClusMOO are 6. The value of SC produced by proposed AMOSHO_CFS is 0.6254. The results reveal that AMOSHO_CFS algorithm outperforms the other competitor approaches in terms of number of clusters and SC.

7. Conclusion

This paper presents a novel astrophysics-based multi-objective approach for automatically finding the clusters and features. The results show that the proposed approach is able to get the effective clusters and features, simultaneously. The proposed approach uses novel concept of dynamic threshold. Efficient searching is made possible by proposing a novel fitness function. The proposed technique is validated on real-life environments. Statistical tests have been carried out to establish the statistical significance of results. The performance of proposed AMOSHO_CFS is also tested on two real-life applications to demonstrate its effectiveness. It has been observed that proposed technique is better than other competitive techniques.

References
2. G. Dhiman and V. Kumar, Knowl.-Based Syst. 159 (2018) 20.
Astrophysics inspired multi-objective approach for automatic clustering