2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
Banff Center, Banff, Canada, October 5-8, 2017

A Novel Clustering Algorithm Based on Fitness Proportionate Sharing

Xuyang Yan, Abdollah Homaifar, Shabnam Nazmi and Mohammad Razeghi-Jahromi
Department of Electrical and Computer Engineering
North Carolina Agricultural and Technical State University
Greensboro, North Carolina
Email: xyan@aggies.ncat.edu, homaifar@ncat.edu, snazmi@aggies.ncat.edu, mrazeghijahromi@ncat.edu

***Abstract*—Existing clustering techniques primarily rely on prior knowledge about the data, such as the number of clusters and radii. However, in real applications, the number of clusters and the radii of clusters are usually unknown. Therefore, the performance of clustering methods with overlapping data is degraded due to their limitations in finding all cluster centers with uneven density values. Hence, a new clustering algorithm based on fitness proportionate sharing is proposed to map the problem into a multimodal optimization problem. In this paper, clusters are considered as niches, and the individuals with the highest density values of each niche are the cluster centers. Instead of using the traditional sharing strategy, the fitness proportionate sharing strategy is implemented in the identification of niche maxima to overcome the sensitivity of uneven density values of cluster centers. A procedure of niche expansion is employed for the merging of clusters. Simulation results and complexity analysis reveal that the proposed clustering algorithm based on fitness proportionate sharing provides a higher accuracy performance without any prior information.**

I. Introduction

As an unsupervised learning technique, clustering is widely used in the statistical analysis for extracting information based on the characteristics of data. The objective of clustering is to maximize the similarity between data samples within the same cluster and maximize the dissimilarity between clusters. Generally, clustering is categorized as prototype-based and non-prototype-based. Prototype-based clustering is most commonly used in real applications. Examples are K-means, subtractive clustering and fuzzy C-means. The proposed algorithm in this paper is a prototype-based clustering algorithm. Most of the existing prototype-based clustering algorithms usually require prior knowledge and have high computational complexities, which is not readily applicable to most real world problems. For example, the K-means algorithm usually requires a fixed number of clusters and it may trap at local optimums when its initial clusters centers are not selected properly. The subtractive clustering is an improved version of mountain clustering [1]. It has a significant superiority in avoiding being trapped in local optimums by removing the effect of identified cluster centers on the potential values of the remaining data samples in the next iteration. However, it uses the density of data samples as the potential value for selecting cluster centers one by one until it reaches the threshold value. To select a proper threshold value, prior information about the data samples is required. This requires a lot of computations because calculations are needed to update potential values after each cluster identification. Similar to the K-means clustering algorithm, fuzzy C-means (FCM) [2] introduces the concept of membership into the clustering algorithm. With a membership matrix, the data sample can have a level of belongingness to all of the clusters, and centers are updated based on their membership values. A cost function is used to determine the termination or continuation of the procedure and a proper threshold value is required to produce a more accurate result. Specifically, a smaller threshold value will provide better performance while increasing the computational complexity.

Considering the deficiency of existing clustering algorithms without prior information, many recent studies proposed that clustering performance can be improved by the application of evolutionary techniques. The genetic algorithm (GA) [3] is the most commonly used evolutionary technique in clustering algorithms. Examples include the genetic K-means algorithm (GKA) [4], a genetic algorithm with re-arrangement for K-means clustering (GAGR) [5] and genetic algorithm based clustering technique (GA-clustering) [6]. An improved K-means clustering methodology by genetic niching [7] was introduced by Sheng in 2004, in which GA was applied as a searching technique for locating optimal cluster centers. In that research, k cluster centers are arranged into a single individual and each one is a candidate set of cluster centers. The fitness value of each individual is inversely proportional to the summation of the distances within each cluster, and the best individual has the smallest summation of the distance. Deterministic crowding niching is used to perform the GA, evolve the population to search for better cluster centers and mediate the sensitivity of K-means clustering algorithm to initial cluster centers. The principle behind deterministic crowding is to replace the parent with the most similar offspring when the offspring has a higher fitness value than the parent. However, it still relies on prior knowledge of the data to find a proper k, which limits its application in real life.

With the dynamic niching clustering technique proposed by Gan and Warwick [8] in solving mutilmodal optimization problems, the author developed a robust dynamic niching genetic algorithm with niche migration for automatic clustering problem (DNNM-clustering) [9] in order to automatically

Corresponding Author: A. Homaifar, Telephone: (336) 2853271

978-1-5386-1645-1/17/$31.00 ©2017 IEEE

identify optimal cluster centers. It considers each individual as a candidate set of cluster centers and N individuals are selected from the data set to construct its population set. The dynamic niche identification procedure is activated in each generation to divide the population into different niches, then a merge of niches is implemented to eliminate unnecessary niches. The traditional fitness sharing strategy in GA with niching is used to find all maximas in a given multimodal function. It is very similar to the population migration between cities. The average resource will be decrease when the current city is too crowded and people will migrate to other cities. However, as in most multimodal problems, traditional fitness sharing will cause the loss of peaks or clusters because it can only find the peak or cluster with the highest fitness, which will degrade the performance of clustering.

As discussed above, the main challenges of traditional clustering problems lie in their strong dependency on prior knowledge and computational complexity. Usually, no prior information about data is available in real life and a high cost of computation is required to approximate actual clusters of data. Hence a new clustering algorithm based on fitness proportionate sharing (FPS-clustering) is proposed in this paper. Similar to the DNNM-clustering algorithm in [9], this new algorithm uses a modified dynamic niche identification to identify all of the peaks based on their fitness, and applies a new fitness proportionate sharing [10] strategy to drive population migration between niches. The merging of niches will happen after all niches are identified and it will be repeated until the niches are stable. In this new algorithm, the GA does not participate in the search process of cluster centers and no certain number of generations is required.

The rest of this paper is organized as follows. Section II discusses the details of the proposed clustering algorithm and introduces a new fitness sharing strategy for the identification of clusters. Simulation results and a performance comparison between three commonly used clustering algorithms and the proposed algorithm are presented in section III. Finally, the conclusion and future work are discussed in section IV.

II. Methodology

In this section, a novel clustering algorithm based on fitness proportionate sharing (FPS-clustering) is discussed in detail.

A. Fitness Function

As discussed before, the objective of optimal clustering is to discover a set of cluster centers which have a minimum inner-distance and maximum inter-distance simultaneously. With this purpose, a candidate set of cluster centers are randomly selected from a given data set, and the proposed algorithm will identify optimal cluster centers based on their strength. To measure the strength of the individual, a density based objective function defined in [11] and [12] is used to represent the fitness of individuals in this study. Generally, euclidean distance is most commonly used in the objective function to reflect the similarity between data samples and it can be expressed as following:

$$f(x_i) = \sum_{j=1}^{n} \left(e^{-\frac{\|x_i - xj\|^2}{\beta}} \right)^{\gamma}. \quad (1)$$

In Eq. (1), n is the total number of data samples and β is the variance of the data set. It is clear that the fitness value for each individual is inversely proportionate to the distance from itself to the rest of the points, which indicates that the dense data point will have a higher fitness while sparse data points have a lower fitness. The value of γ will affect the density estimation of the data distribution. To find proper estimates of this parameter for different data sets used in this study, a correlation comparison algorithm [11] is employed and the resulting values for γ are listed in Table I. Data set 1 has nine overlapped clusters with uneven density values while data set 2 has nine non-overlapped clusters with even density distribution. Data set 3 is a high dimensional data with three overlapped clusters.

TABLE I
ESTIMATE VALUE FOR γ FOR SIMULATED DATA SETS

Estimated	Data set 1	Data set 2	Data set 3
γ	20	15	5

B. Fitness Proportionate Sharing

In traditional sharing strategy [13], the raw fitness $f(x_{ij})_{old}$ of individual j in the niche i is divided by the niche count m_i and the shared fitness $f(x_{ij})_{new}$ can be obtained by following Eq. (2) and (3):

$$f(x_{ij})_{new} = \frac{f(x_{ij})_{old}}{m_i}, \quad (2)$$

and

$$m_i = \sum_{j=1}^{n_i} sh(d_{ij}), \quad (3)$$

where n_i is the number of individuals in i^{th} niche and shared function $sh(d_{ij})$ is usually expressed as:

$$sh(d_{ij}) = 1 - \left(\frac{d_{ij}}{\sigma_{sh}} \right)^{\alpha_{sh}}, \quad (4)$$

where σ_{sh} is initial radius of niches and α_{sh} is a constant parameter which accounts for the shape of the shared function. α_{sh} is usually set to equal 1 and it yields a triangular shape of the sharing function. The issue with this traditional sharing policy comes from its deficiency in finding all optimal clusters when they have uneven density values. It can only find the cluster with the highest density and lose some clusters when the density of clusters are unevenly distributed. Hence the proposed algorithm aims to use fitness proportionate sharing to identify local maxima for each niche and then refines the niches by merging small niches into larger groups. Fitness proportionate sharing firstly appeared as a niching technique in [10] and [14]. It was developed to avoid the sensitivity of

unequal maximas in local niches. In the niche identification procedure, the raw fitness $f(x_{ij})_{old}$ of individual j within niche i will be scaled to shared fitness $f(x_{ij})_{new}$ by Eq. (5):

$$f(x_{ij})_{new} = \frac{f(x_{ij})_{old}}{\sum_{j=1}^{n_i} f(x_{ij})_{old}}. \quad (5)$$

After updating the fitness with the represented sharing scheme, the difference between niche maximas will be mediated, which offers an equal significance for all niches regardless of their peak values. With this new sharing strategy, all of the peaks will be treated fairly and those peaks with smaller fitness values will be able to be identified, which can guarantee all optimal clusters to be discovered.

C. Niche Expansion with Merging

With the sharing strategy proposed in the previous subsection, a set of niches can be discovered with an initial niche radius σ_{sh}. The identification of niches is very similar to a procedure of population migration [10]. The resource of each individual within the current niche will be shared based on the proposed sharing strategy, and the average resource will decrease. Then, individuals will have a strong potential to move to other niches with lower densities. Driven by this mechanism, all potential niches can be discovered. However, some identified niches may be very close to each other and a new niche can be formed by merging them together. Therefore, a procedure of niche expansion with merging used in DNMM-clustering [9] is implemented by checking the communication between niches to refine niches. The communication between niches is determined by checking whether a valley exists between two peaks of niches. The "valley" [9] here refers to a point between two peaks that has a fitness value lower than the minimum of those two peaks and it reflects the existence of a boundary between clusters. A merge will happen when two niches communicate with each other. The value of m is not a fixed number and it can improve the accuracy when a large value is chosen. In this paper, it is set to be 100. By merging communicating niches, initial niches can finally evolve to optimal clusters. The details of the expansion of niches with merging is explained in algorithm 1.

Algorithm 1 Niche Expansion with Merging

A list of merging niches: $Mergelist = \emptyset$
Check for Valley: $Com = \emptyset$
Niches that was checked for merge: $marked = \emptyset$
for $i = 1 : numofniches$ **do**
  if i is not marked **then**
    Select the closest peak to current peak i as the neighbor peak and record its index Ni
    Linearly generate m points between two neighboring points and calculate their fitness value f_m
    $FN \leftarrow$ fitness of the neighbor peak
    $FP \leftarrow$ fitness of current peak
    for $k = 1 : m$ **do**
      if $f(k) < Min(FP, FN)$ **then**
        $Merge = 0$
        Exit the loop
      end if
    end for
    if Merge **then**
      $Com = [i, Ni]$;
      $marked = [marked, Ni]$;
    end if
  end if
  $Mergelist = [Mergelist, Com]$
  $marked = [marked, i]$
end for
Do the merge for niches on the mergelist and the refined niches will be returned to the main function.

D. FPS Clustering Algorithm

With a given data set, a candidate set of N data samples will be randomly selected from the test data set. The FPS-clustering algorithm applies fitness proportionate sharing to discover potential local maximals of niches and then performs niche identification with an initial radius σ_{sh}. The value of initial radius σ_{sh} and N will affect both the performance of clustering and computational complexities. Specifically, a larger value of N with a smaller value of σ_{sh} can provide a better quality of clustering performance but bring more computations. Later, a merge among all identified niches based on the communication [9] between niches will be activated, which allows the proposed algorithm to find final optimal niches. Thus, FPS-clustering does not need any predefined number of clusters or radii and it can evolve to the final optimal clustering results automatically. The entire procedure of the FPS-clustering algorithm is summarized in algorithm 2.

III. EXPERIMENT AND RESULTS

In this section, three artificial numeric data sets are used to compare the performance of three well-known clustering algorithms (fuzzy C-means, subtractive and K means) with the proposed method (FPS-clustering). The details of each data set are described in the simulation results and discussion section. 20% of each data set are randomly selected to form its candidate set in the simulation and the initial niche radius is set to 1% of the spread d_{max} of the data set. Usually, the value of the size of the candidate set and niche radius are not fixed and they can be adjusted based on the data sets. To reduce the effect of randomness in the initial candidate set selection, the proposed algorithm was repeated by 30 times and an average of its performance was calculated. Since the remaining methods are deterministic, they are simulated just once. In the end, a computational complexity analysis is presented.

A. Performance Metric

To evaluate the performance of the proposed clustering algorithm, two statistical score functions, including overall accuracy (OA) and kappa index (KI) as proposed in [16], are used in this paper to provide a comprehensive comparison. These two parameters are based on the confusion matrix that

Algorithm 2 FPS-clustering

$Maxlimit \leftarrow$ The maximum of data sample
$Minlimit \leftarrow$ The minimum of data sample
$d_{max} \leftarrow$ The spread between $Maxlimit$ and $Minlimit$
$r \leftarrow$ The percentage for initial radius takes in d_{max}
$Popsize = N, \sigma_{sh} = r \times d_{max}$;
Initialize the candidate set with N samples from the data set.
Calculate the fitness value for the candidate set based on Eq. (1).
while Not all individuals of candidate set are assigned **do**
    Select the individual with highest fitness as the i^{th} peak;
    $Niche = \emptyset$;
    for all $j = 1 : N$ **do**
        Compute the distance d from individual j to i^{th} peak;
        if $d \leq \sigma_{sh}$ **then**
            $Assign \leftarrow$ Individulas assigned to niche i
        end if
    end for
    Do the fitness proportionate sharing to update the fitness of samples within current niche i ;
    $i = i + 1$;
end while
while Niches are changing **do**
    Do the Merge among all niches;
end while
The final peaks will be the optimal cluster centers.

reflects the consistency between the original data clusters and the test results. Assume n to be the total number of data samples in the data space, and n_{ii} to be the diagonal element of the confusion matrix that defines the number of points that are assigned to cluster i that originally belonged to cluster i. The number of samples belonging to the original cluster i can be obtained by finding the sum of each row of the confusion matrix: $n_{row_i} = \sum_j n_{ij}$. And the number of samples that are classified into cluster j will be the sum of each column in the confusion matrix: $n_{col_j} = \sum_i n_{ij}$. The following are the equations for the performance indexes used in the study:

1) Overall accuracy

$$OA = \frac{\sum_i n_{ii}}{n}. \tag{6}$$

2) Kappa index

$$KI = \frac{n \times \sum_i n_{ii} - \sum_i n_{row_i} \times n_{col_i}}{n^2 - \sum_i n_{row_i} \times n_{col_i}}. \tag{7}$$

B. Simulation Data and Results

Data set 1 is a randomly generated data set with 3300 samples, which lie around nine clusters that are close to each other. 660 samples will be selected from this data set to initialize the candidate set. OA and KI are employed to calculate the accuracy of training using the data set and presented in Tables II and III. As shown in Fig. 1, the proposed algorithm can successfully discover all of the nine clusters without specifying the cluster numbers and cluster radii. The remaining methods can also find all nine clusters correctly except for the subtractive clustering algorithm when a proper number of clusters and radii are provided. From Tables II and III, the proposed FPS-clustering algorithm has a better performance than K-means, subtractive and fuzzy C-means. Based on those two parameters defined previously, the performance of subtractive clustering algorithm for data set 1 is much worse than the other algorithms because of overlapping between different clusters.

Data set 2 is a randomly generated data set of 2700 samples in nine clusters. These nine clusters are separate from one another and each cluster has 300 samples evenly distributed around it. A candidate set of 540 data samples are randomly selected from this data set. As it can be seen in Fig. 2, all clustering algorithms were able to find all nine clusters while the proposed FPS-clustering algorithm has better performance based on its overall accuracy and kappa index according to Tables II and III.

Data set 3 is an artificial data set which comes from the UCI machine learning repository. It includes 210 data samples with seven features [15]. The candidate set for this data set consists of 42 samples that are randomly selected from this data set. Due to its high dimensionality, the final clustering result is not shown in this paper. Instead, the overall accuracy and kappa index of each method are calculated and shown in Tables II and III. It is clear that K-means, fuzzy C-means and subtractive clustering do not perform as well as FPS-clustering, which shows superiority of proposed algorithm in handling overlapping, high-dimensional data. The subtractive clustering and K-means will fail to find correct cluster centers if the radius or initial clusters are not selected appropriately.

TABLE II
OVERALL ACCURACY FOR SIMULATION RESULT

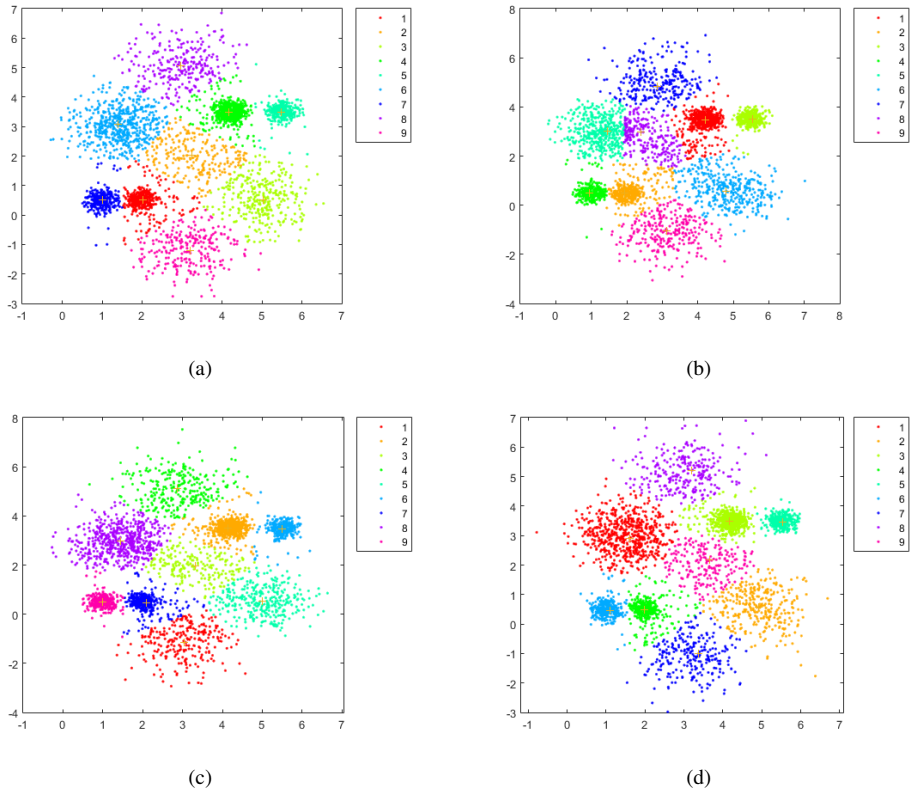
OA	fuzzy C-means	subtractive	K-means	FPS-clustering
Data set 1	0.905	0.761	0.907	0.917
Data set 2	0.988	0.987	0.984	0.9893
Data set 3	0.862	0.776	0.862	0.871

TABLE III
KAPPA INDEX FOR SIMULATION RESULT

KI	fuzzy C-means	subtractive	K-means	FPS-clustering
Data set 1	0.913	0.791	0.918	0.932
Data set 2	0.987	0.987	0.982	0.9879
Data set 3	0.792	0.664	0.793	0.807

C. Complexity Analysis

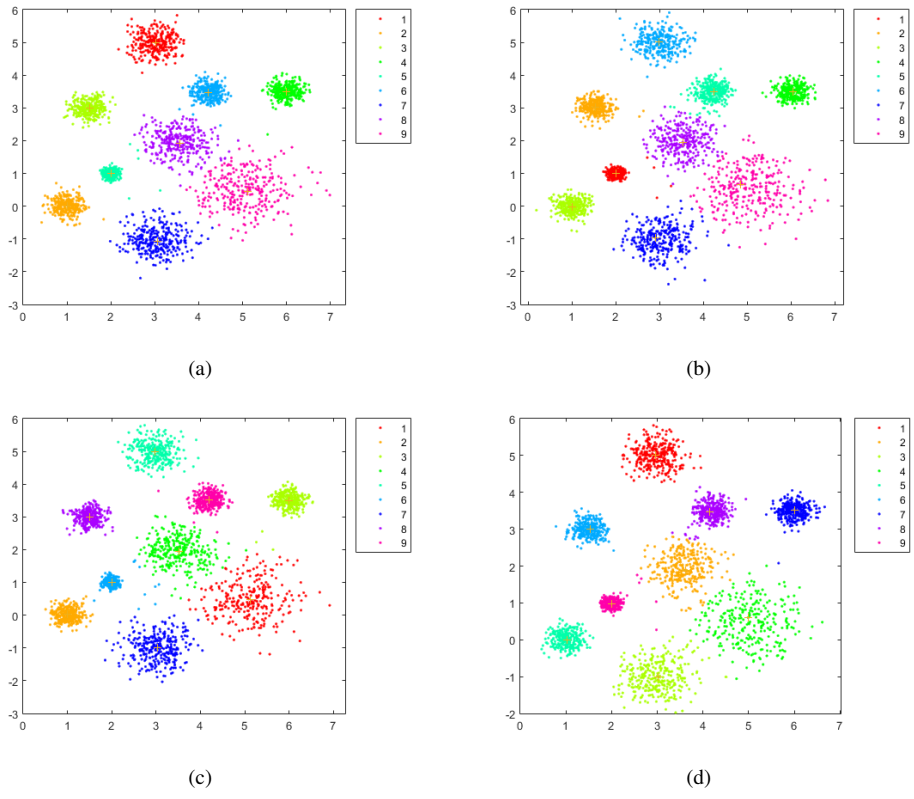
With respect to computational complexity, each candidate samples requires a number of $n - 1$ distance calculations for



(a) (b)

(c) (d)

Fig. 1. Simulation results for (a): fuzzy C-means, (b): subtractive, (c): K-means and (d): FPS-clustering from data set 1.



(a) (b)

(c) (d)

Fig. 2. Simulation results for (a): fuzzy C-means, (b): subtractive, (c): K-means and (d): FPS-clustering from data set 2.

computing their fitness, and a set of N candidate samples from n data points will have $N \times (n-1)$ distance calculations in the beginning. Then for expansions between niches, every merge will involve $100 \times (n-1)$ calculations as 100 points are generated between two neighboring niche maximas, and p merges will take $p \times 100 \times (n-1)$ calculations. Hence, the total number of calculations of the proposed algorithm is $(N + p \times 100) \times (n-1)$.

IV. Conclusion

In this paper, a novel clustering algorithm based on fitness proportionate sharing (FPS-clustering) was developed to data clustering without a predefined cluster number and radii. This new algorithm uses fitness proportionate sharing as a major mechanism for cluster evolution through the dynamic niche identification and implements an automatic merging of niches. This new algorithm has many real-life applications because it can provide high quality results without any prior knowledge of data. With the fitness proportionate sharing strategy, the effects of uneven peak values can be mediated and all peaks will have an equal significance in the optimization procedure. From the simulation results, it is clear that the proposed algorithm has a better performance than other algorithms because of its higher accuracy, especially for overlapping data, which shows its superiority to other methods.

Streaming data will play a more significant role in many areas like real-time object tracking, time series prediction and autonomous control for robotics. The partition of streaming data with clustering should be considered for information collection and system structure modeling by dividing the complex problems into several smaller parts, which can be easily solved. Therefore, further study about improving the proposed algorithm for the dynamic clustering problems is needed.

Acknowledgment

This paper is based on research sponsored by Air Force Research Laboratory and OSD under agreement number FA8750-15-2-0116. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory and OSD or the U.S. Government. The authors would like to thank Air Force Research Laboratory and Office of the Secretary of Defense (OSD).

References

[1] R. R. Yager and D. P. Filev, "Approximate clustering via the mountain method," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 8, pp. 1279–1284, 1994.

[2] J. C. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," 1973.

[3] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine learning*, vol. 3, no. 2, pp. 95–99, 1988.

[4] K. Krishna and M. N. Murty, "Genetic k-means algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 3, pp. 433–439, 1999.

[5] D.-X. Chang, X.-D. Zhang, and C.-W. Zheng, "A genetic algorithm with gene rearrangement for k-means clustering," *Pattern Recognition*, vol. 42, no. 7, pp. 1210–1222, 2009.

[6] U. Maulik and S. Bandyopadhyay, "Genetic algorithm-based clustering technique," *Pattern recognition*, vol. 33, no. 9, pp. 1455–1465, 2000.

[7] W. Sheng, A. Tucker, and X. Liu, "Clustering with niching genetic k-means algorithm," in *Genetic and Evolutionary Computation–GECCO 2004*. Springer, 2004, pp. 162–173.

[8] J. Gan and K. Warwick, "A genetic algorithm with dynamic niche clustering for multimodal function optimisation," in *Artificial Neural Nets and Genetic Algorithms*. Springer, 1999, pp. 248–255.

[9] D.-X. Chang, X.-D. Zhang, C.-W. Zheng, and D.-M. Zhang, "A robust dynamic niching genetic algorithm with niche migration for automatic clustering problem," *Pattern recognition*, vol. 43, no. 4, pp. 1346–1360, 2010.

[10] A. Workineh and A. Homaifar, "Fitness proportionate niching: Maintaining diversity in a rugged fitness landscape," in *Proceedings of the International Conference on Genetic and Evolutionary Methods (GEM)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2012, p. 1.

[11] M.-S. Yang and K.-L. Wu, "A similarity-based robust clustering method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 4, pp. 434–448, 2004.

[12] L. A. Zadeh, "Similarity relations and fuzzy orderings," *Information sciences*, vol. 3, no. 2, pp. 177–200, 1971.

[13] D. E. Goldberg, J. Richardson *et al.*, "Genetic algorithms with sharing for multimodal function optimization," in *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*. Hillsdale, NJ: Lawrence Erlbaum, 1987, pp. 41–49.

[14] A. Workineh and A. Homaifar, "A fitness proportionate reward sharing: a viable default hierarchy formation strategy in lcs," in *Proceedings of the International Conference on Genetic and Evolutionary Methods (GEM)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2012, p. 1.

[15] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[16] D. Chang, Y. Zhao, L. Liu, and C. Zheng, "A dynamic niching clustering algorithm based on individual-connectedness and its application to color image segmentation," *Pattern Recognition*, vol. 60, pp. 334–347, 2016.